



TESIS TE142599

**PERANCANGAN SISTEM PENGATURAN KECEPATAN
MOTOR INDUKSI 3 PHASA DENGAN *DIRECT TORQUE
CONTROL* (DTC) MENGGUNAKAN *SLIDING MODE
CONTROL* (SMC) BERBASIS ALGORITMA GENETIKA**

M. Nur Faizi

2213202002

DOSEN PEMBIMBING :

Dr. Ir. Mochammad Rameli

Eka Iskandar, ST., MT.

PROGRAM MAGISTER

BIDANG KEAHLIAN SISTEM PENGATURAN

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA

2016



THESIS TE-142599

**DESIGN FOR SPEED CONTROL SYSTEM IN 3 PHASE
INDUCTION MOTOR WITH DIRECT TORQUE CONTROL
(DTC) USING SLIDING MODE CONTROL (SMC) BASED
ON GENETIC ALGORITHM**

M. Nur Faizi

2213202002

SUPERVISOR

Dr. Ir. Mochammad Rameli

Eka Iskandar, ST., MT.

**POST GRADUATE PROGRAM
DEPARTMENT OF ELECTRICAL ENGINEERING
AREAS OF CONTROL SYSTEMS
FACULTY OF INDUSTRIAL OF TECHNOLOGY
SEPULUH NOPEMBER OF TECHNOLOGY
SURABAYA
2016**


**Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)
Di
Institut Teknologi Sepuluh Nopember**


**Oleh :
M. Nur Faizi
NRP: 2213202002**


**Tanggal Ujian : 28 Januari 2016
Periode Wisuda : Maret 2016**

Disetujui oleh:


1. Dr. Ir. Mochammad Rameli. (Pembimbing I)
NIP: 1954 1227 1981 03 1002


2. Eka Iskandar, S.T., M.T. (Pembimbing II)
NIP: 1980 0528 2008 12 1001


3. Dr. Trihastuti Agustinah, S.T., M.T. (Penguji)
NIP: 1968 0812 1994 03 2001


4. Ir. Katjuk Astrowulan, MS. EE. (Penguji)
NIP: 1951 0415 1978 03 1001


5. Ir. Ali Fatoni, M.T. (Penguji)
NIP: 1962 0603 1989 03 1002



Direktur Program Pascasarjana,

Prof. Ir. Djauhar Manfaat. M. Sc, Ph. D
NIP: 1960 1202 1987 01 1001

PERANCANGAN SISTEM PENGATURAN KECEPATAN MOTOR INDUKSI 3 PHASA DENGAN *DIRECT TORQUE CONTROL* (DTC) MENGUNAKAN *SLIDING MODE CONTROL* (SMC) BERBASIS *ALGORITMA GENETIKA*

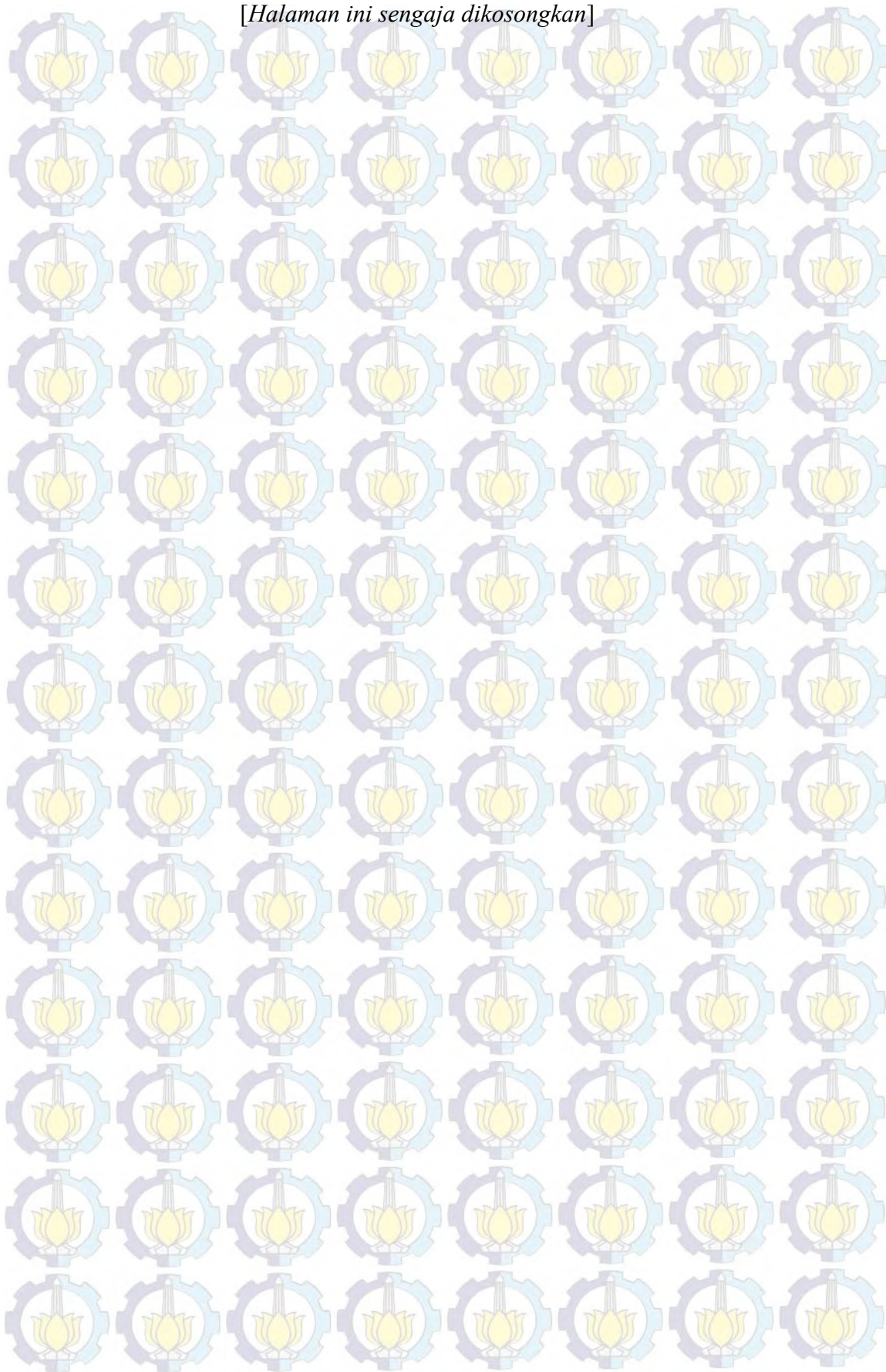
| | |
|----------------|----------------------------|
| Nama Mahasiswa | : M. Nur Faizi |
| NRP | : 2213202002 |
| Pembimbing | : Dr. Ir. Mochammad Rameli |
| Co-Pembimbing | : Eka Iskandar, ST., MT |

ABSTRAK

Secara umum dalam dunia industri, motor induksi lebih banyak digunakan daripada motor arus searah. Dikarenakan sifat motor induksi yang kokoh, handal, mudah dalam perawatan, dan harga relatif murah. Tetapi pengontrolan pada motor induksi lebih kompleks dibandingkan dengan motor arus searah, hal ini disebabkan oleh kompleksitas dinamika motor induksi, sehingga algoritma pengaturannya lebih kompleks. Berdasarkan masalah tersebut penulis melakukan penelitian pada motor induksi menggunakan metode *Sliding Mode Control* (SMC) berbasis *Algoritma Genetika*, di mana *Algoritma Genetika* dirancang untuk masalah optimalisasi parameter *Sliding Mode Control* (SMC) yaitu (*gain K*) yang optimal, guna untuk mengatasi kekurangan pada *Direct Torque Control* (DTC) konvensional yang masih menimbulkan fluktuasi *ripple* fluks dan fluktuasi *ripple* torsi yang tinggi pada saat kondisi *steady state*, sehingga dari tujuan metode yang diusulkan, hasil dari sistem yang dirancang dapat mengatur kecepatan putar motor induksi sesuai referensi yang diberikan sebesar 1000 rpm dengan *settling time* yaitu 0.6282 detik, dan juga dapat meminimalkan fluktuasi *ripple* fluks dan fluktuasi *ripple* torsi.

Kata kunci: *Direct Torque Control*, *Sliding Mode Control*, Motor Induksi, *Algoritma Genetika*, *gain K*.

[Halaman ini sengaja dikosongkan]



Design for Speed Control System in 3 Phase Induction Motor with Direct Torque Control (DTC) using Sliding Mode Control (SMC) Based on Genetic Algorithm

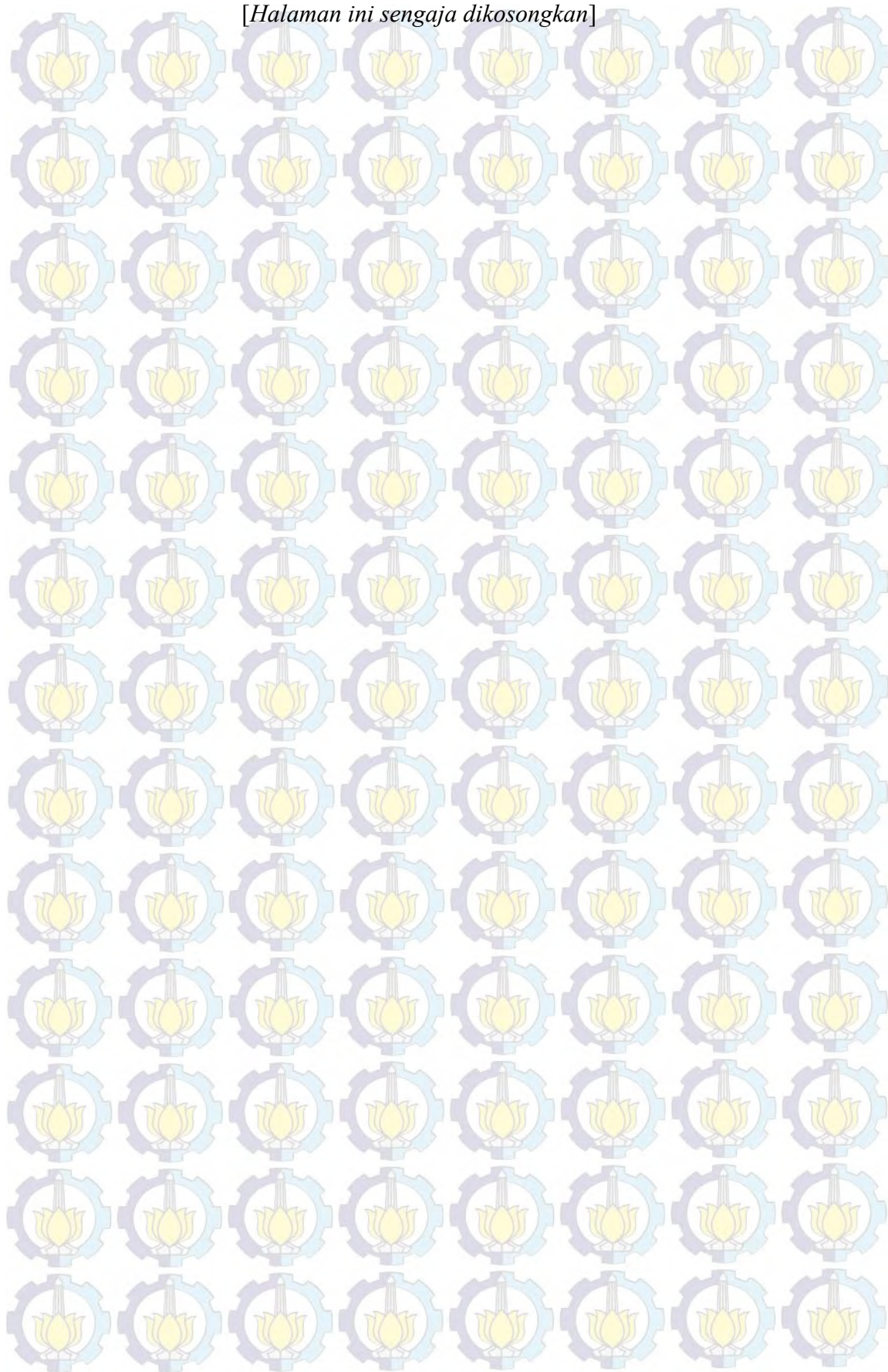
By : M. Nur Faizi
Student Identity Number : 2213202002
Supervisor : Dr. Ir. Mochammad Rameli
Co-Supervisor : Eka Iskandar, ST., MT

ABSTRACT

Generally in the industry, the induction motor is more widely used than the direct current motor. Due to the characteristic of induction motor that are sturdy, reliable, easy in maintenance, and the prices are relatively cheap, but control system in induction motor is more complicated than the direct current motor, the problem is caused by the complexity of the induction motor dynamics, and it needs complex algorithm to solve. Based on the problems, the author try to do research in induction motor by using Sliding Mode Control (SMC) methods with Genetic Algorithm. Genetic Algorithm is designed to optimized the parameter of Sliding Mode Control (SMC), namely (gain K) is optimal, in order to overcome the shortage in conventional Direct Torque Control (DTC), which is still causing high-ripple flux's fluctuation dan high-ripple torque's fluctuation during in steady state, the result from the design system with the methode that is proposed, can control the speed rotation of induction motor according to the references given at 1000 rpm the settling time is 0,6282 seconds, and can minimize the ripple flux's fluctuation and the ripple torque's fluctuation.

Key words: Direct Torque Control, Sliding Mode Control, Induction Motor, Genetic Algorithm, gain K.

[Halaman ini sengaja dikosongkan]



KATA PENGANTAR

Bismillaahirrahmaanirraahiim

Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT yang senantiasa selalu memberikan ridho dan rahmat-Nya. Tak lupa shalawat serta salam penulis haturkan kepada Nabi besar junjungan kita Rasulullah Muhammad SAW, sehingga penulis dapat menyelesaikan Tesis ini dengan judul:

**Perancangan Sistem Pengaturan Kecepatan Motor Induksi 3 Phasa Dengan
Direct Torque Control (DTC) Menggunakan *Sliding Mode Control (SMC)*
Berbasis *Algoritma Genetika***

Dalam menyelesaikan Tesis ini, penulis berpegang pada teori yang pernah diperoleh dan bimbingan dari para dosen pembimbing Tesis. Selain itu, dengan adanya bantuan dan dukungan pihak-pihak lain, sehingga Tesis ini dapat terselesaikan. Oleh karena itu, pada kesempatan ini penulis hendak menyampaikan ucapan terima kasih dan teriring doa semoga Allah SWT membalas setiap kebaikan dengan kebaikan yang berlipat ganda kepada semua keluarga yang telah memberikan dorongan moral, doa serta semangat untukku dalam mengerjakan Tesis ini. Secara khusus penulis mengucapkan termakasih sebesar-besarnya kepada:

1. Allah SWT, Subhanallah, Alhamdulillah, Wa Laa ilaa ha Illallah, Allahu Akbar.
2. Bapak Mahali, dan Ibu Suprapti, selaku kedua orang tua penulis. Beliauulah ayah dan ibu tercinta atas segala kasih sayang dan doanya selama ini.
3. Bapak Dr. Ir. Mochammad Rameli selaku dosen pembimbing 1, Bapak Eka Iskandar, ST., MT selaku dosen pembimbing 2, dan Bapak Ir. Katjuk Astrowulan, MSEE selaku dosen wali.
4. Seluruh keluarga besarku atas doa restunya padaku.

5. Bapak/Ibu dosen Bidang Studi Sistem Pengaturan yang telah dengan ikhlas dan tulus memberikan materi ajarnya kepada penulis, Allah SWT yang akan membalas semua kebaikan Bapak/Ibu. Aamiin.
6. Teman-teman mahasiswa S1 dan S2, serta teman-teman laboratorium A202 yang selalu memberikan masukan kepada penulis.
7. Semua pihak yang tidak dapat disebutkan satu per satu yang telah memberi bantuan baik secara langsung maupun tidak langsung.

Semoga tulisan ini tidak hanya berarti di dunia namun juga membawa manfaat di akhirat dan dicatat sebagai amal ibadah disisi Allah SWT, sehingga pada akhirnya buku Tesis ini mampu menambah pengetahuan dan pemahaman pada bidang ilmu kontrol bagi penulis serta untuk rekan-rekan mahasiswa Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Surabaya, Februari 2016

Penulis

DAFTAR ISI

| | |
|--|-------------|
| JUDUL | i |
| LEMBAR PENGESAHAN | iii |
| PERNYATAAN KEASLIAN TESIS | v |
| ABSTRAK | vii |
| ABSTRAK | ix |
| KATA PENGANTAR | xi |
| DAFTAR ISI | xiii |
| DAFTAR GAMBAR | xv |
| DAFTAR TABEL | xvii |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan | 3 |
| 1.5 Kontribusi | 4 |
| 1.6 Metodologi Penelitian | 4 |
| BAB II KAJIAN PUSTAKA DAN DASAR TEORI | 7 |
| 2.1 Kajian Pustaka | 7 |
| 2.2 Motor Induksi 3 Phasa | 8 |
| 2.2.1 Konstruksi | 9 |
| 2.2.2 Prinsip Kerja Motor Induksi 3 Phasa | 11 |
| 2.2.3 Model Matematika Motor Induksi | 12 |
| 2.2.4 Model Dinamik Beban Motor Induksi | 23 |
| 2.3 <i>Direct Torque Control (DTC)</i> | 24 |
| 2.3.1 Prinsip Dasar DTC | 25 |
| 2.3.2 Kontroler Hiterisis | 27 |
| 2.3.3 Estimator dan Sektor Fluks Stator | 28 |
| 2.3.4 Tabel <i>Switching</i> | 29 |
| 2.3.5 Inverter Sumber Tegangan | 31 |
| 2.4 <i>Sliding Mode Control (SMC)</i> | 33 |
| 2.5 Algoritma Genetika | 37 |
| 2.5.1 Aplikasi Algoritma genetika | 38 |

| | |
|--|----|
| 2.5.2 Sistem Operasi Algoritma Genetika | 40 |
| BAB III PERANCANGAN SISTEM | 48 |
| 3.1 Gambaran Umum Sistem | 47 |
| 3.2 Perancangan Simulasi Model Motor Induksi | 49 |
| 3.3 Perancangan DTC Pada Simulasi Model Motor Induksi | 51 |
| 3.3.1 Perancangan Kontroler Histerisis | 52 |
| 3.3.2 Perancangan Estimator | 53 |
| 3.3.3 Perancangan Sektor Sudut Fluks Stator | 54 |
| 3.3.4 Perancangan Tabel <i>Switching</i> | 55 |
| 3.4 Perancangan Model Motor Induksi Dengan DTC-PI-FLC | 57 |
| 3.5 Perancangan Model Motor Induksi Dengan DTC-PI-SMC | 58 |
| 3.6 Perancangan Kontroler <i>Slidng Mode Control</i> (SMC) | 59 |
| 3.7 Perancangan Kontroler SMC-AG Pada <i>Plant</i> Motor Induksi | 63 |
| BAB IV HASIL DAN PEMBAHASAN | 65 |
| 4.1 Simulasi Model Motor Induksi | 65 |
| 4.2 Simulasi Motor Induksi Dengan DTC-PI-FLC | 67 |
| 4.3 Simulasi Motor Induksi Dengan DTC-PI-SMC | 71 |
| 4.4 Simulasi Motor Induksi Dengan DTC-SMC-AG | 75 |
| 4.5 Simulasi perbandingan Respon Kecepatan Motor Induksi | 79 |
| BAB V PENUTUP | 87 |
| 5.1 Kesimpulan | 87 |
| 5.2 Saran | 88 |
| DAFTAR PUSTAKA | 89 |
| LAMPIRAN | 91 |
| RIWAYAT PENULIS | 99 |

DAFTAR GAMBAR

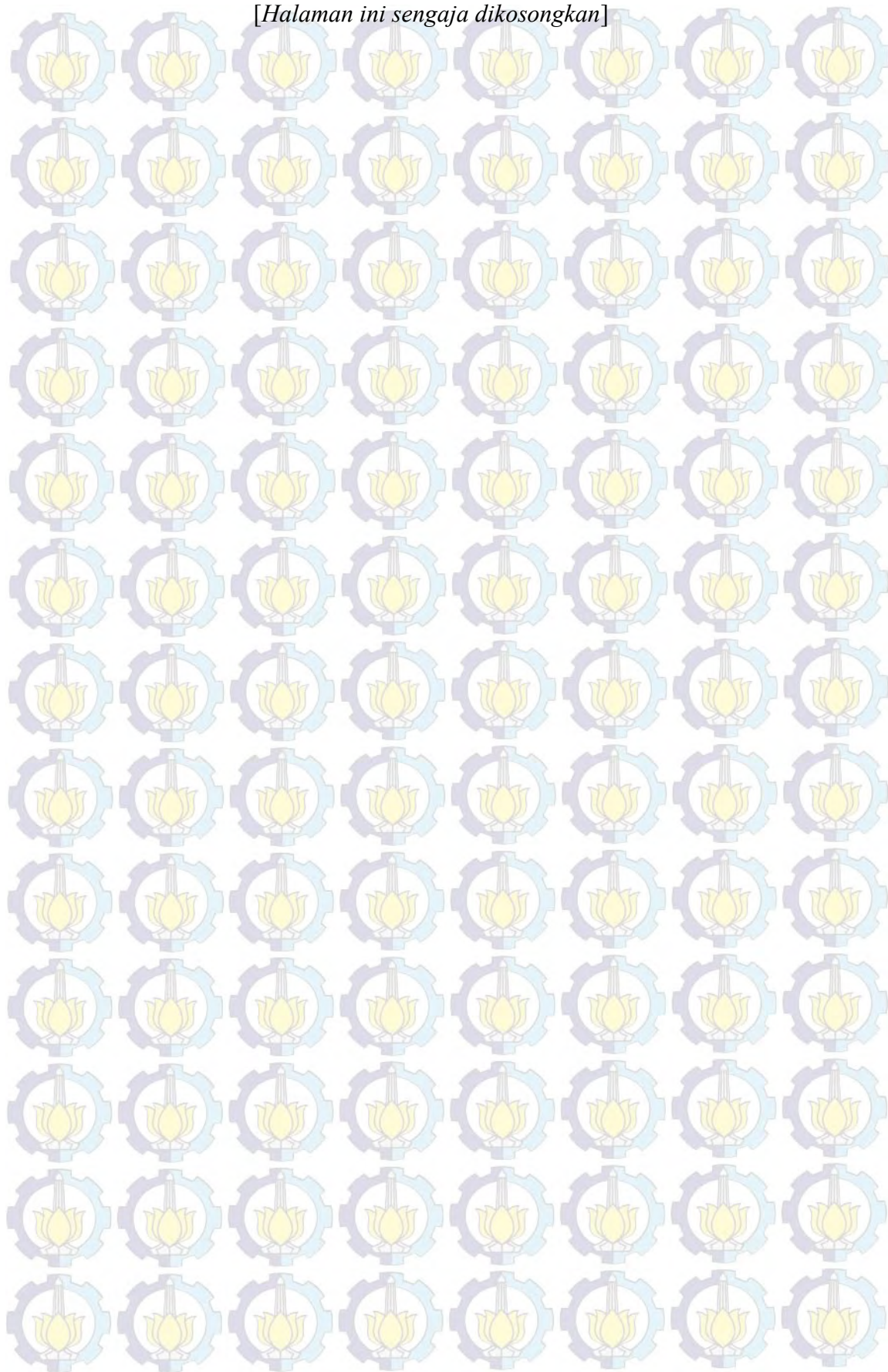
| | |
|--|----|
| Gambar 2.1 Arus 3 Phasa..... | 8 |
| Gambar 2.2 Konstruksi Motor Induksi 3 Phasa..... | 9 |
| Gambar 2.3 Konstruksi Rotor Lilit..... | 10 |
| Gambar 2.4 Konstruksi Motor Sangkar..... | 11 |
| Gambar 2.5 (a) Efek Kopling Pada Kumputan Stator dan Rotor 3 Phasa..... | 13 |
| Gambar 2.5 (b) Kumputan 2 Phasa..... | 13 |
| Gambar 2.6 Transformasi Clarke..... | 14 |
| Gambar 2.7 Transformasi Park..... | 15 |
| Gambar 2.8 Rangkaian Ekuivalen Stator $\alpha\beta$ dan d-q..... | 17 |
| Gambar 2.9 Rangkaian Ekuivalen Rotor $\alpha\beta$ dan d-q..... | 18 |
| Gambar 2.10 Rangkaian Ekuivalen Motor Induksi 3 Phasa Sumbu d-q..... | 19 |
| Gambar 2.11 Model Dinamik Beban Inersa dan Damper..... | 23 |
| Gambar 2.12 Diagram Blok Sistem DTC Motor Induksi Tiga Phasa..... | 24 |
| Gambar 2.13 Pembagian Ruang Vektor Tegangan Bidang d-q..... | 26 |
| Gambar 2.14 Vektor Tegangan Saat Fluks Stator..... | 27 |
| Gambar 2.15 Kontroler Kisterisis Fluks dan Torsi..... | 28 |
| Gambar 2.16 Diagram Blok Estimator DTC..... | 28 |
| Gambar 2.17 Vektor Tegangan Inverter dan Sektor <i>Switching</i> Fluks Stator..... | 31 |
| Gambar 2.18 Rangkaian Inverter 3 phasa..... | 31 |
| Gambar 2.19 Urutan <i>Switching</i> Inverter 3 Phasa..... | 32 |
| Gambar 2.20 Diagram Kontrol Modus Luncur..... | 34 |
| Gambar 2.21 Diagram Phasa Trajektori Status..... | 36 |
| Gambar 2.22 Struktur Umum Algoritma Genetika..... | 39 |
| Gambar 2.23 Proses Mutasi Pada Algoritma Genetika..... | 44 |
| Gambar 3.1. Diagram Alir Perancangan Sistem Keseluruhan..... | 47 |
| Gambar 3.2 Diagram Blok Sistem Keseluruhan..... | 48 |
| Gambar 3.3 Perancangan Simulasi Model Motor Induksi..... | 51 |
| Gambar 3.4 Perancangan Kontroler Histerisis Fluks dan Torsi DTC..... | 52 |
| Gambar 3.5 Perancangan Estimator DTC..... | 53 |
| Gambar 3.6 Transformasi Tegangan dari abc ke dq..... | 54 |
| Gambar 3.7 Perancangan Sektor Sudut Fluks..... | 55 |
| Gambar 3.8 Perancangan Tabel <i>Switching</i> | 56 |
| Gambar 3.9 Simulasi Motor Induksi DTC-PI Menggunakan FLC..... | 57 |
| Gambar 3.10 Simulasi Motor Induksi DTC-PI Menggunakan SMC..... | 58 |

| | |
|--|----|
| Gambar 3.11 Diagram Blok Kontroler <i>Sliding Mode Control</i> (SMC)..... | 61 |
| Gambar 3.12 Perancangan Model Kontroler <i>Sliding Mode Control</i> (SMC)..... | 62 |
| Gambar 3.13 Diagram Blok Optimasi Algoritma Genetika Pada SMC..... | 62 |
| Gambar 3.14 Model Simulink Optimasi Menggunakan Algoritma Genetika... | 63 |
| Gambar 3.15 Perancangan Kontroler SMC <i>Algoritma Genetika</i> | 64 |
| Gambar 4.1 Respon Kecepatan Rotor Model Motor Induksi Tanpa Beban..... | 65 |
| Gambar 4.2 Respon Torsi Elektromagnetik Model Motor Tanpa Beban..... | 66 |
| Gambar 4.3 Respon Kecepatan Rotor Model Motor Induksi Berbeban..... | 66 |
| Gambar 4.4 Respon Torsi Elektromagnetik Model Motor Berbeban | 67 |
| Gambar 4.5 Respon Kecepatan Rotor DTC-PI Menggunakan FLC | 68 |
| Gambar 4.6 Respon Torsi Elektromagnetik DTC-PI Menggunakan FLC | 69 |
| Gambar 4.7 Respon Fluks Stator DTC-PI Menggunakan FLC | 70 |
| Gambar 4.8 Respon Arus Stator DTC-PI Menggunakan FLC | 70 |
| Gambar 4.9 Respon Kecepatan Rotor DTC-PI Menggunakan SMC | 72 |
| Gambar 4.10 Respon Torsi Elektromagnetik DTC-PI Menggunakan SMC | 73 |
| Gambar 4.11 Respon Fluks Stator DTC-PI Menggunakan SMC..... | 74 |
| Gambar 4.12 Respon Arus Stator DTC-PI Menggunakan SMC..... | 75 |
| Gambar 4.13 Respon Kecepatan Rotor DTC-PI Menggunakan SMC | 76 |
| Gambar 4.14 Respon Torsi Elektromagnetik DTC-PI Menggunakan SMC | 77 |
| Gambar 4.15 Respon Fluks Stator DTC-PI Menggunakan SMC..... | 78 |
| Gambar 4.16 Respon Arus Stator DTC-PI Menggunakan SMC..... | 79 |
| Gambar 4.17 Respon Kecepatan dan Torsi DTC Referensi 1000rpm | 80 |
| Gambar 4.18 Respon Fluks dan Arus Stator DTC Referensi 1000rpm..... | 81 |
| Gambar 4.19 Respon Kecepatan dan Torsi DTC Referensi 1100rpm | 82 |
| Gambar 4.20 Respon Fluks dan Arus Stator DTC Referensi 1100rpm..... | 83 |
| Gambar 4.21 Respon Kecepatan dan Torsi DTC Referensi 1200rpm | 84 |
| Gambar 4.22 Respon Fluks dan Arus Stator DTC Referensi 1200rpm..... | 85 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Pengaruh Vektor Tegangan terhadap Nilai Fluks Stator | 27 |
| Tabel 2.2 Tabel Switching dari Vektor Tegangan Inverter..... | 30 |
| Tabel 2.3 Kondisi <i>Switching</i> Inverter..... | 33 |
| Tabel 2.4 Skema <i>Binary Encoding</i> | 41 |
| Tabel 3.1 Parameter Motor Induksi 3 HP | 50 |
| Tabel 3.2 Sektor Sudut Fluks Stator | 54 |
| Tabel 4.1 Perbandingan Karakteristik Respon Waktu | 86 |

[Halaman ini sengaja dikosongkan]



BAB I

PENDAHULUAN

1.1 Latar Belakang

Secara umum, dalam dunia industri motor arus bolak-balik lebih banyak digunakan daripada motor arus searah. Dikarenakan sifatnya yang kokoh, handal, mudah dalam perawatan, dan harga relatif murah [1], sedangkan motor arus searah banyak digunakan sebagai alternatif pada mesin-mesin yang memerlukan pengaturan kecepatan tertentu karena memiliki karakteristik dinamis yang linier dibanding motor arus bolak-balik. Namun motor arus searah juga memiliki beberapa kelemahan yaitu dari segi ekonomis karena biaya pemeliharaannya lebih mahal dibanding motor arus bolak-balik, sedangkan pengontrolan pada motor arus bolak-balik jauh lebih kompleks dibandingkan dengan motor arus searah karena disebabkan oleh kompleksitas dinamika motor induksi, sehingga algoritma pengaturannya lebih kompleks [2].

Usaha-usaha memperbaiki keterbatasan motor arus bolak-balik tersebut untuk keperluan pengontrolan terdiri dari dua metode yaitu pengaturan skalar dan vektor. Pengaturan skalar beroperasi pada keadaan *steady state* dan juga dibutuhkan pengaturan kecepatan sudut dari arus, tegangan, dan *linkage* fluks dalam *space vector*. Dengan penjelasan demikian bahwa pengaturan skalar tidak dapat beroperasi dalam keadaan *transient state*. Untuk mengatasi permasalahan tersebut dapat menggunakan pengaturan vektor, di mana metode tersebut bisa bekerja dalam keadaan *transient state* dan tidak hanya dapat mengatur sudut kecepatan dan magnitudo, tapi juga arus, tegangan, dan fluks. Salah satu metode pengaturan yang populer untuk pengaturan vektor adalah *Field Oriented Control* (FOC) yang diperkenalkan oleh F.Blaschke (Direct FOC) dan Hasse (Indirect FOC) diawal tahun 1970, di mana FOC dapat memberikan performa dan efisiensi tinggi untuk berbagai aplikasi industri [3].

FOC bagus untuk menghasilkan performa dinamik tinggi, *ripple* torsi dan fluks yang rendah, tapi FOC memiliki kekurangan, seperti memerlukan pengaturan arus, membutuhkan 2 koordianat transformasi dan sensitifitas parameter mesin yang tinggi. Kekurangan ini dapat dieliminasi menggunakan

Direct Torque Control (DTC) yang diusulkan oleh Isao Takahashi dan Toshihiko Noguchi, pada tahun 1980. Dibandingkan dengan FOC, DTC lebih sederhana dalam hal struktur, kebutuhan komputasi yang sedikit, efisiensi dan performa yang tinggi [3].

Direct Torque Control (DTC) merupakan salah satu skema pengaturan berdasarkan pengaturan fluks stator dan torsi yang memberikan respon cepat dan kokoh yang diimplementasikan pada motor arus bolak-balik. DTC ini memiliki kelebihan antara lain lebih sederhana dan mempunyai performa dinamik yang baik serta tidak sensitif terhadap perubahan parameter, khususnya tahanan stator. Akan tetapi penggunaan DTC konvensional memiliki beberapa kekurangan yaitu dapat menimbulkan fluktuasi *ripple* fluks dan torsi yang tinggi saat kondisi *steady state* [3].

Dalam hal untuk mengatasi masalah ini, diusulkan DTC dengan *Sliding Mode Control* (SMC) di mana *Sliding Mode Control* (SMC) adalah salah satu teknik pengontrolan yang memiliki sifat kokoh terhadap gangguan maupun perubahan parameter [5]. Skema kontrolnya adalah dengan memaksa *vector error* dan *delta error (trajektori)* menuju kearah permukaan luncur. Namun kekurangan dari penggunaan kontroler *Sliding Mode Control* (SMC) adalah adanya fenomena *chattering*. Munculnya fenomena *chattering* ini, dikarenakan proses pemeliharaan *trajektori* pada permukaan luncur tersebut. Fenomena *chattering* ini dapat mengganggu stabilitas dari sistem kontrol yang kurang baik untuk respon sistem terutama pada fluktuasi *ripple* fluks dan torsi sehingga keberadaannya perlu direduksi.

1.2 Rumusan Masalah

Dari penjelasan latar belakang di atas, bahwa permasalahan sistem pengaturan kecepatan motor induksi 3 fasa bergantung kepada respon fluktuasi *ripple* fluks dan torsi yang dibangkitkan, semakin besar fluktuasi *ripple* yang dibangkitkan, dapat mempengaruhi stabilitas putaran rotor motor induksi 3 fasa. Dengan digunakan kontroler *Sliding Mode Control* (SMC), sistem memiliki kemampuan yang kokoh terhadap gangguan dan perubahan parameter, namun permasalahan *Sliding Mode Control* (SMC) adalah penentuan parameter (*gain K*)

yang dilakukan dengan cara coba-coba (dilakukan dengan cara memilih nilai yang sesuai atau nilai yang memenuhi kriteria). Oleh karena itu, pada tesis ini akan dirancang sistem pengaturan kecepatan motor induksi 3 phasa dengan *Sliding Mode Control* berbasis *Algoritma Genetika*, dimana *Algoritma Genetika* dirancang untuk mendeteksi nilai *ripple* torsi, sehingga hasil dari optimasi *Algoritma Genetika* adalah nilai yang terbaik atau yang optimal dan digunakan untuk *Tuning* (*gain K*) pada SMC untuk mendapatkan nilai *ripple* yang kecil.

1.3 Batasan Masalah

Penulis membatasi permasalahan yang akan dibahas pada penelitian ini nantinya. Berdasarkan hasil membaca literatur, maka penulis membatasi permasalahan tersebut sebagai berikut:

1. *Plant* yang dikendalikan merupakan motor induksi 3 phasa dalam bentuk model d-q.
2. Efek dari rugi-rugi besi dan tembaga diabaikan.
3. Ruang lingkup yang dibahas yaitu permasalahan respon kecepatan pada saat kondisi *transient*, kemampuan sistem dalam mereduksi fluktuasi *ripple* fluks dan torsi.
4. Optimalisasi parameter *Sliding Mode Control* (SMC) dilakukan secara *off-line*, menggunakan metode *Algoritma Genetika*.
5. Parameter *plant* tetap.

1.4 Tujuan

Sistem pengaturan yang dirancang diharapkan mampu menghasilkan kecepatan putaran motor yang diinginkan sesuai dengan referensi yang diberikan dan *Algoritma Genetika* bisa menghasilkan (*gain K*) yang optimal untuk parameter *Sliding Mode Control* (SMC). Dengan demikian diharapkan dapat memberikan hasil yang baik dalam usaha mengatur kecepatan motor induksi 3 phasa dalam hal meminimalkan fluktuasi *ripple* fluks dan torsi terutama pada saat *steady state*.

1.5 Kontribusi

Dapat memberikan sumbangan pemikiran dan referensi akan sebuah metode kontrol yang mana dalam penelitian ini digunakan *Sliding Mode Control* (SMC) berbasis *Algoritma Genetika* yang diharapkan mampu men-*Tuning* parameter SMC yaitu *gain K* dan dapat meminimalkan fluktuasi *ripple* fluks dan torsi, sehingga pengontrolan kecepatan putar motor induksi 3 phasa dapat bekerja dengan baik.

1.6 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini antara lain :

1. Studi Literatur

Tahap ini diperlukan sebagai langkah awal sebelum mengerjakan penelitian. Hal-hal yang dipelajari antara lain mempelajari cara kerja *plant*, cara menggunakan *plant*, dan cara mengontrol *plant* melalui buku, jurnal, media *online*, dan lain-lain.

2. Pemodelan Sistem

Pada tahap ini akan dilakukan pemodelan fisik sistem pengaturan motor induksi 3 phasa.

3. Perancangan Sistem

Berdasarkan hasil pemodelan sistem. Estimasi kecepatan motor akan dibandingkan dengan kecepatan referensi untuk menghasilkan error. *Error* dan *delta error* kecepatan rotor sebagai masukan pada *Sliding Mode Control* (SMC), *Algoritma genetika* digunakan untuk masalah optimalisasi parameter *Sliding Mode Control* (SMC). Perancangan sistem ini akan dilakukan dengan perangkat lunak MATLAB.

4. Pengujian Dan Analisa

Pada tahap pengujian dan analisis, sistem yang dibangun khusus pada kondisi berbeban, respon kecepatan diharapkan stabil dan dapat memenuhi dengan nilai referensi yang diberikan, dan respon fluktuasi *ripple* fluks dan torsi dapat minimum.

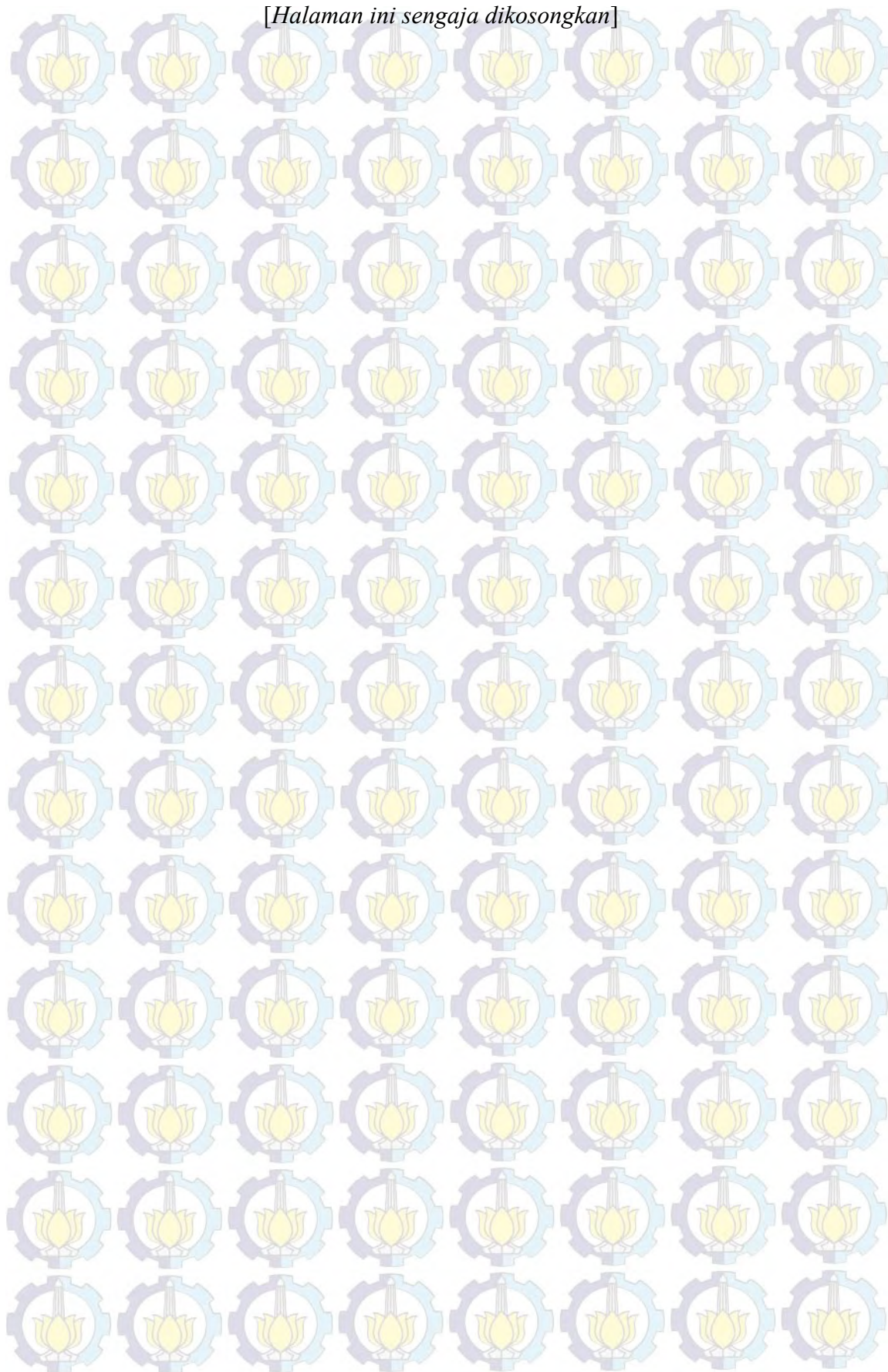
5. Kesimpulan

Kesimpulan diperoleh sesuai dengan hasil pengujian dan analisis yang dilakukan.

6. Penulisan Laporan Tesis

Penulisan buku Tesis dimulai dari penelitian sampai tahap akhir. Buku Tesis ditulis secara intensif bila proses analisis dan pengujian telah selesai.

[Halaman ini sengaja dikosongkan]



BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Kajian Pustaka

Beberapa peneliti telah melakukan penelitian yang berhubungan dengan sistem pengaturan motor induksi 3 fasa. Dalam penelitian ini, diperlukan teori-teori atau metode-metode yang berhubungan dengan apa yang diteliti. Teori atau metode yang berhubungan di antaranya adalah tentang pengaturan fluktuasi torsi.

Penelitian yang dilakukan T. Ramesh, dan A Kumar dalam penelitiannya menyebutkan, metode konvensional *Direct Torque Control* (DTC) mempunyai skema pengaturan yang sederhana, membutuhkan perhitungan yang sedikit, performa yang tinggi, dan efisien, tetapi akan penggunaan DTC ini menimbulkan fluktuasi *ripple* fluks dan torsi yang tinggi [3]. Sedangkan Chien-Feng Hu, Rong-Bin Hong, dan Chang-Huan Liu dalam penelitian pengaturan kecepatan dengan tuning kontroler PI, di mana tuning kontroler PI dapat mengatur kecepatan putaran motor induksi mencapai referensi kecepatan yang diinginkan, tetapi membutuhkan *continues tuning* nilai yang pasti dari gain proporsional (K_p) dan integral (K_i) untuk memperoleh performa kontroler yang bagus [4].

Dalam penelitian mengenai pengaturan kecepatan putaran motor induksi, Tanvir Ahammad, Abdul R. Beig, Khalifa Al-Hosani didalam penelitiannya [5], meningkatkan *Direct Torque Control* dari Motor Induksi dengan pendekatan modifikasi *Sliding Mode Control* (SMC) dapat membuktikan bahwa pengaturan motor induksi dengan metode *Sliding Mode Control* (SMC) kecepatan motor memiliki respon yang unggul, dan ketahanan yang baik dalam menghadapi ketidakpastian termasuk gangguan beban. Selain itu, akurasi *performance* dapat dicapai, serta fluktuasi *ripple* fluks dan torsi berkurang jika dibandingkan dengan teknik DTC konvensional. Namun pada penelitian ini permasalahan *chattering* belum bisa diatasi dengan sepenuhnya dimana fenomena *chattering* ini timbul akibat proses pemeliharaan trayektori pada salah satu sinyal kontrol dari metode *Sliding Mode Control* (SMC).

Maka untuk mengatasi permasalahan tersebut, pada penelitian ini dilakukan pengaturan kecepatan putaran motor induksi 3 fasa dengan *Direct Torque Control*

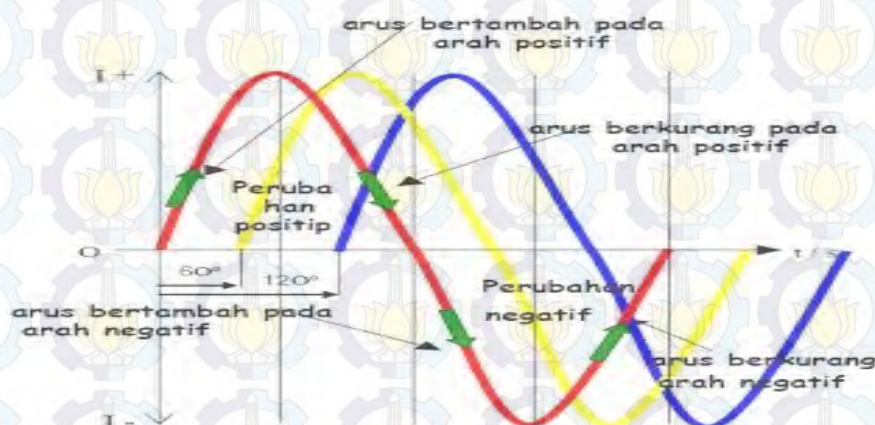
(DTC) Menggunakan *Sliding Mode Control* (SMC) Berbasis *Algoritma Genetika*. Dimana *Algoritma Genetika* mampu mengatasi permasalahan optimasi kombinasi, yaitu mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai kemungkinan banyak solusi [13].

2.2 Motor Induksi 3 Phasa [8]

Motor induksi adalah suatu mesin listrik yang merubah energi listrik menjadi energi gerak dengan menggunakan gandingan medan listrik dan mempunyai slip antara medan stator dan medan rotor. Motor induksi merupakan motor yang paling banyak kita jumpai dalam industri.

Tipe sangkar tupai merupakan tipe yang paling banyak digunakan di dunia industri. Mesin tipe ini sangat kuat dan sederhana. Menurut penemuan seorang yang ahli di bidangnya pada akhir abad ke 19, bahwa mesin tipe ini masih bertahan dan tidak tertandingi popularitasnya dalam penggunaan di industri. Jenis motor induksi yang lebih sedikit digunakan adalah jenis *wound rotor*. Jenis motor tipe ini biasanya digunakan dalam aplikasi khusus, biasanya yang berhubungan dengan keberadaan dan kemudahan untuk mengakses lilitan rotor.

Motor induksi 3 phasa bekerja dengan memanfaatkan perbedaan fasa sumber untuk menimbulkan gaya putar pada rotornya. Jika pada motor AC 1 phasa untuk menghasilkan beda fasa diperlukan penambahan komponen Kapasitor, pada motor induksi 3 phasa perbedaan fasa sudah didapat langsung dari sumber seperti terlihat pada gambar 2.1.

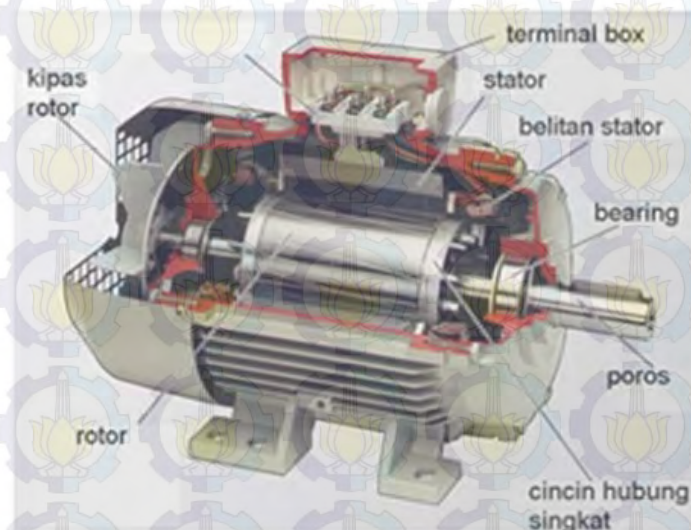


Gambar 2.1 Arus 3 phasa [http://wiskeydeoch.blogspot.com/2011]

Pada Gambar 2.1. arus 3 phasa memiliki perbedaan phase 60 derajat antar phasanya. Dengan perbedaan ini, maka penambahan kapasitor tidak diperlukan.

2.2.1 Kontruksi

Motor induksi 3 phasa memiliki dua komponen dasar yaitu stator dan rotor, bagian rotor dipisahkan dengan bagian stator oleh celah udara yang sempit (air gap) dengan jarak antara 0,4 mm sampai 4 mm. Tipe dari motor induksi 3 phasa berdasarkan lilitan pada rotor dibagi menjadi dua macam yaitu rotor belitan (*wound rotor*) adalah tipe motor induksi yang memiliki rotor terbuat dari lilitan yang sama dengan lilitan statornya dan rotor sangkar tupai (*Squirrel-cage rotor*) yaitu tipe motor induksi di mana konstruksi rotor tersusun oleh beberapa batangan logam yang dimasukkan melewati slot-slot yang ada pada rotor motor induksi, kemudian setiap bagian disatukan oleh cincin sehingga membuat batangan logam terhubung singkat dengan batangan logam yang lain. Gambar motor induksi 3 phasa dapat dilihat pada Gambar 2.2.

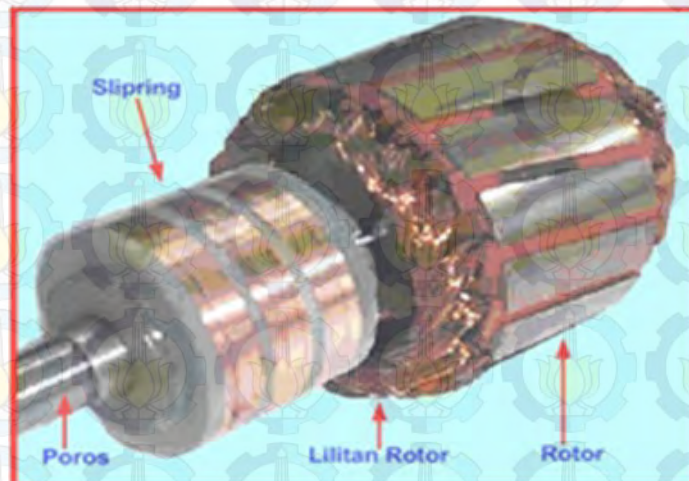


Gambar 2.2 Kontruksi motor induksi 3 phasa [circuitbooks.wordpress.com]

Motor induksi sering disebut motor asinkron (tidak serempak), disebut demikian karena jumlah putaran rotor tidak sama dengan putaran medan magnet stator. Jenis rotor yang digunakan yaitu:

- Rotor lilit (*wound rotor*)

Motor jenis ini berkapasitas besar, juga sering disebut motor slipring atau motor cincin seretatau cincin hubung singkat. Rotor jenis ini memiliki belitan–belitan kawat jika didistribusikan maka motor jenis ini juga dapat kita fungsikan sebagai alternator (generator) dengan demikian pada rotor ini akan memiliki kutub–kutub pada stator belitan internal rotor dari motor ini dihubungkan secara bintang (tiga fasa) kemudian terminal belitan tersebut dikeluarkan dan disambungkan ke tiga buah slip ring terisolasi yang diletakkan pada poros motor dengan sikat diatasnya. Ketiga sikat ini secara eksternal dihubungkan ke suatu reostat yang membentuk bintang. Reostat pada motor ini berfungsi untuk meningkatkan torsi asut motor pada saat periode pengusutan. Apabila motor ini bekerja pada kondisi normal, maka slip ring secara otomatis terhubung pendek. Sehingga ring diatas tangkai terhubung bersama oleh suatu logam yang tertekan selanjutnya secara otomatis sikat tersebut terangkat dari slip ring yang berfungsi untuk mengurangi rugi–rugi gesekan. Kontruksi rotor lilit dapat dilihat pada Gambar 2.3.

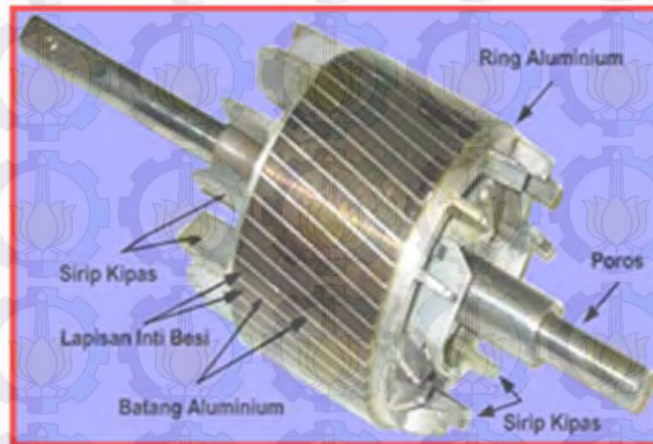


Gambar 2.3 Konstruksi Rotor Lilit [circuitbooks.wordpress.com]

- Rotor sangkar (*square cage rotor*)

Motor jenis ini sering disebut motor dengan rotor hubung singkat. Secara umum hampir 90% dari motor induksi banyak menggunakan rotor dengan jenis

ini. Karena rotor jenis ini, pada motor induksi adalah paling sederhana dan kuat rotor jenis ini dibuat dari baja silikon dan terdiri dari inti yang berbentuk silinder yang sejajar dengan alur/slot dan diisi dengan tembaga atau alumunium yang berbentuk batangan. Kontruksi rotor sangkar dapat dilihat pada Gambar 2.4.



Gambar 2.4 Konstruksi Rotor Sangkar [circuitbooks.wordpress.com]

2.2.2 Prinsip Kerja Motor Induksi 3 Fasa

Prinsip kerja pada motor induksi erat kaitannya dengan Hukum Faraday dan Gaya Lorentz. Hukum Faraday menjelaskan mengenai konduktor yang memotong garis-garis gaya dari suatu medan magnet (fluks) yang konstan, maka pada konduktor tersebut akan timbul tegangan induksi. Sedangkan Gaya Lorentz yang berlaku pada prinsip kerja motor induksi menjelaskan bahwa apabila konduktor yang dialiri arus berada pada medan magnet, maka pada konduktor tersebut akan timbul suatu gaya.

Secara garis besar prinsip kerja dari motor induksi tiga fasa dapat dijelaskan dalam tahapan-tahapan di bawah ini:

- a. Pada kumparan stator yang disuplai sumber tegangan tiga fasa akan timbul medan magnet. Adanya perubahan tegangan (sinusoidal) maka medan magnet akan berubah dan menimbulkan medan putar dengan kecepatan sinkron (N_s). Besarnya nilai kecepatan sinkron dapat dirumuskan sebagai berikut:

$$N_s = \frac{60.f}{p} \quad (2.2)$$

di mana,

N_s = kecepatan sinkron (rpm)

f = frekuensi stator pada motor induksi (Hz)

p = jumlah pasang kutub/poles.

- b. Medan putar pada stator akan memotong konduktor yang ada pada rotor, sehingga akan menimbulkan tegangan induksi (Hukum Faraday).
- c. Tegangan induksi yang timbul pada konduktor yang dihubungkan singkat akan menghasilkan arus. Arus mengalir keluar dari konduktor yang persis berada di bawah permukaan kutub magnet menuju ujung batang konduktor dan kembali melalui konduktor yang lain.
- d. Konduktor yang mengalirkan arus berada di dalam medan magnet dari magnet permanen, maka pada konduktor akan bekerja satu gaya mekanik (Gaya Lorentz). Gaya yang dibangkitkan selalu mendorong konduktor searah dengan arah pergerakan medan magnet.

Arus stator dan arus rotor yang dihasilkan akan menimbulkan torsi elektromagnetik (T_e) pada motor. Jika torsi mula yang dihasilkan cukup besar, maka rotor akan berputar searah dengan medan putar stator. Tegangan terinduksi terjadi jika adanya perbedaan antara kecepatan angular dari medan putar stator (N_s) dengan kecepatan putar rotor (N_r). Perbedaan antara N_s dan N_r disebut dengan slip (S), dinyatakan dengan:

$$s = \frac{\Delta N}{N_s} = \frac{\Delta \omega}{\omega_s} = \frac{N_s - N_r}{N_s} = \frac{\omega_s - \omega_r}{\omega_s} \quad (2.3)$$

di mana,

ω_s = kecepatan angular stator (rad/s)

N_r = kecepatan rotor (rpm)

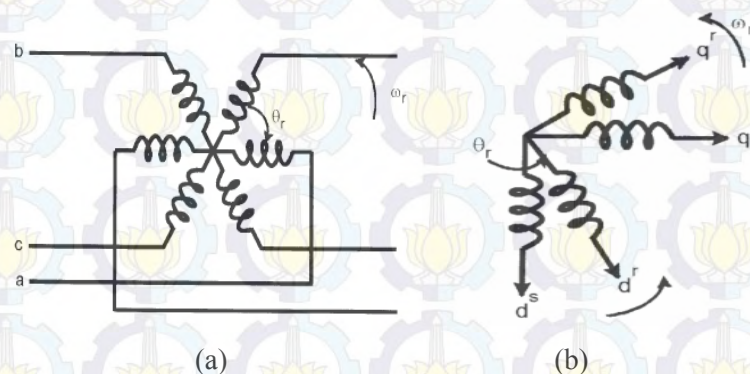
ω_r = kecepatan elektrik rotor (rad/s)

2.2.3 Model Matematika Motor Induksi [11]

Performansi dinamik dari mesin AC agak kompleks karena kumparan rotor 3 fasa bergerak pada lilitan stator 3 fasa. Pada dasarnya kumparan stator dan rotor pada motor induksi dapat dianggap sebagai trafo dengan bagian keduanya

bergerak, di mana koefisien kopling antara fasa stator dan rotor berubah secara kontinyu dengan perubahan dari posisi rotor θ_r . Model dari motor induksi dapat digambarkan dengan persamaan diferensial serta mutual induktansi yang berubah terhadap waktu, tapi model yang lainnya cenderung sangat kompleks. Mesin tiga fasa dapat digambarkan dengan mesin dua fasa yang sebanding. Penggambaran motor induksi dalam dua fasa disimbolkan dalam sumbu d (*direct*) dan q (*quadrature*). Gambar kumparan tiga fasa dan dua fasa dapat dilihat pada Gambar 2.5.

G. Kron mengusulkan transformasi dari variabel stator dan rotor ke frame referensi yang berputar secara sinkron (*synchronously rotating reference frame*) yang bergerak dengan medan magnet yang berputar.



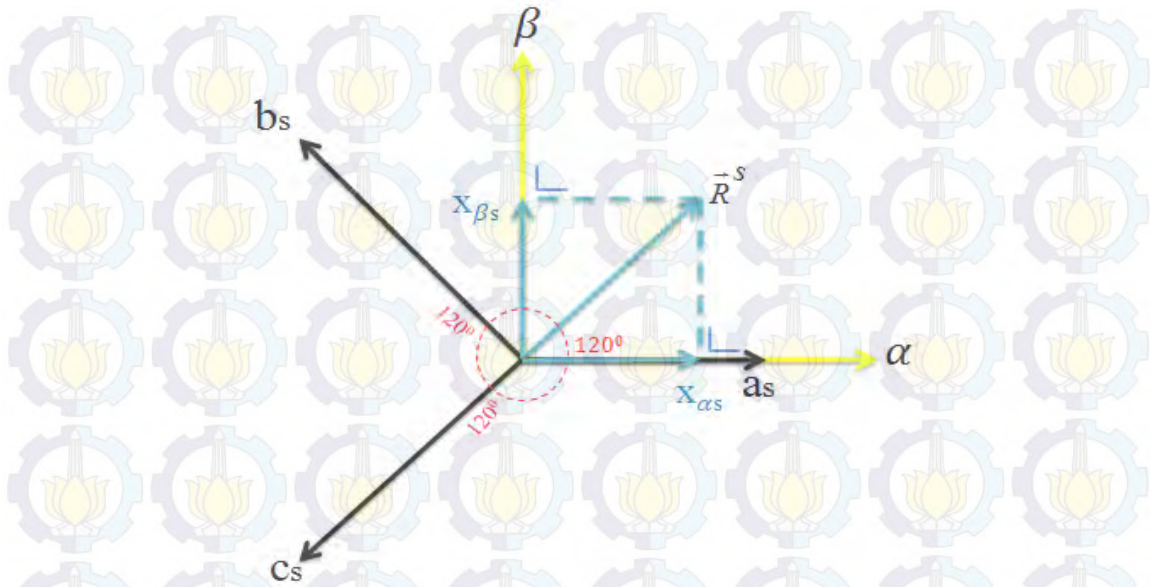
Gambar 2.5 (a) Efek Kopling pada Kumparan Stator dan Rotor Tiga Fasa (b) Kumparan Dua Fasa

Untuk kumparan dua phasa pada Gambar 2.5.(b) diperlukan merepresentasikan rangkaian d_s - q_s dan d_r - q_r dalam frame d_e - q_e yang berputar secara sinkron.

Dalam bentuk sistem koordinat, perubahan sistem koordinat stasioner tiga fasa (a,b,c) menjadi sistem koordinat dua fasa yang berputar (d,q) dipisahkan menjadi dua langkah yaitu Transformasi Clarke dan Transformasi Park:

a. Transformasi Clarke

Transformasi Clarke atau transformasi ($\alpha\beta$) merupakan transformasi tiga fasa (a,b,c) menjadi sistem dua fasa (α,β) yang stasioner. Gambar 2.6. merupakan Gambar transformasi clarke.



Gambar 2.6 Transformasi Clarke

Pada Gambar 2.6. transformasi clarke diatas, vector \vec{R}^s bila dinyatakan dalam bentuk dari komponen tiga fasa (a,b,c):

$$\vec{R}^s = X_{as} \cdot e^{j0^\circ} + X_{bs} \cdot e^{j120^\circ} + X_{cs} \cdot e^{j240^\circ} \quad (2.4)$$

Dengan menggunakan identitas Euler:

$$e^{j\omega t} = \cos(\omega t) + j \sin(\omega t) \quad (2.5)$$

maka komponen X_{as} dan X_{bs} dari vector \vec{R}^s dapat dinyatakan:

$$\begin{aligned} x_{as} + j \cdot x_{bs} &= (\cos(0)X_{as} + \cos(120) \cdot X_{bs} + \cos(240) \cdot X_{cs}) \\ &\quad + j(\sin(0) \cdot X_{as} + \sin(120) \cdot X_{bs} + \sin(240) \cdot X_{cs}) \end{aligned} \quad (2.6)$$

sehingga:

$$\vec{R}^s = \left(X_{as} - \frac{1}{2}X_{bs} - \frac{1}{2}X_{cs} \right) + j \left(\frac{\sqrt{3}}{2}X_{bs} - \frac{\sqrt{3}}{2}X_{cs} \right) \quad (2.7)$$

Untuk mengubah sistem koordinat stasioner tiga fasa (a,b,c) menjadi sistem koordinat dua fasa ($\alpha\beta$). persamaan dalam bentuk matriks dapat ditulis dengan persamaan:

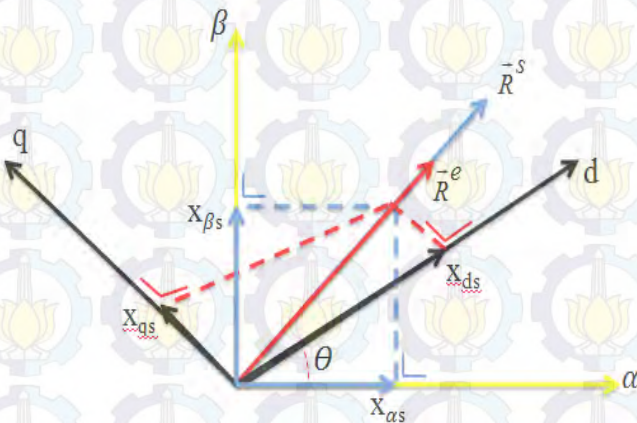
$$\begin{bmatrix} X_{\alpha s} \\ X_{\beta s} \end{bmatrix} = k \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2}\sqrt{3} & -\frac{1}{2}\sqrt{3} \end{bmatrix} \begin{bmatrix} X_{as} \\ X_{bs} \\ X_{cs} \end{bmatrix} \quad (2.8)$$

Dengan k , Transformasi Clarke:

$$k = \frac{2}{3}$$

b. Transformasi Park

Transformasi Park merupakan transformasi sistem dua fasa stasioner (α,β), menjadi sistem dua fasa yang berputar, *direct* dan *quadrature* (dq). Gambar 2.7. merupakan Gambar transformasi park.



Gambar 2.7 Transformasi Park

Dari Gambar 2.7. Transformasi Park, hubungan antara vector \vec{R}^s kerangka referensi stasioner dan vektor \vec{R}^e pada kerangka referensi yang bergerak adalah:

$$\vec{R}^e = \vec{R}^s \cdot e^{-j\theta} \quad (2.9)$$

Dengan menggunakan identitas Euler:

$$e^{j\omega t} = \cos(\omega t) + j \sin(\omega t) \quad (2.10)$$

Persamaan di atas dijabarkan menjadi:

$$\begin{aligned} x_{ds} + j \cdot x_{qs} &= (x_{\alpha s} + j \cdot x_{\beta s}) [\cos(-\theta) + j \sin(-\theta)] \\ x_{ds} + j \cdot x_{qs} &= [x_{\alpha s} \cdot \cos(\theta) + x_{\beta s} \cdot \sin(\theta)] + j [-x_{\alpha s} \cdot \sin(\theta) + x_{\beta s} \cdot \cos(\theta)] \end{aligned} \quad (2.11)$$

Atau bila persamaan di atas dinyatakan dalam matriks, maka:

$$\begin{bmatrix} X_{ds} \\ X_{qs} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X_{\alpha s} \\ X_{\beta s} \end{bmatrix} \quad (2.12)$$

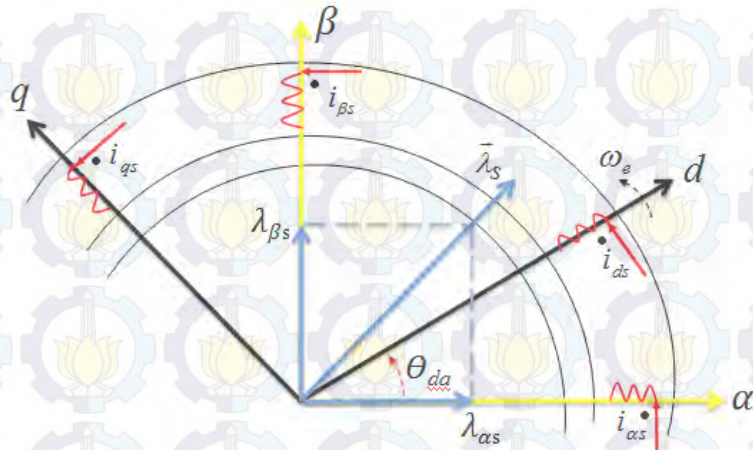
Dari sistem koordinat yang mengubah sistem koordinat stasioner tiga fasa (a,b,c), menjadi sistem koordinat dua fasa yang berputar (dq), maka didapat persamaan fluks stator dan rotor:

$$\begin{aligned} \lambda_{qs} &= L_s \cdot i_{qs} + L_m \cdot i_{qr} \\ \lambda_{ds} &= L_s \cdot i_{ds} + L_m \cdot i_{dr} \\ \lambda_{qr} &= L_r \cdot i_{qr} + L_m \cdot i_{qs} \\ \lambda_{dr} &= L_r \cdot i_{dr} + L_m \cdot i_{ds} \end{aligned} \quad (2.13)$$

Kemudian untuk mencari persamaan tegangan d-q, kumparan stator dapat ditunjukkan dengan persamaan berikut ini:

$$\begin{aligned} V_{\alpha s} &= R_s \cdot i_{\alpha s} + \frac{d}{dt} \lambda_{\alpha s} \\ V_{\beta s} &= R_s \cdot i_{\beta s} + \frac{d}{dt} \lambda_{\beta s} \end{aligned} \quad (2.14)$$

Rangkaian kumparan ekuivalen stator $\alpha\beta$ dan d-q dapat dilihat pada Gambar 2.8.



Gambar 2.8 Rangkaian Ekuivalen stator $\alpha\beta$ dan d-q

Dari Gambar 2.8. dapat dilihat untuk mencari tegangan stator d-q yaitu dengan cara mengkombinasikan kedua persamaan di atas dapat ditulis:

$$\vec{V}_{s_ \alpha\beta}^{\alpha} = R_s \vec{i}_{s_ \alpha\beta}^{\alpha} + \frac{d}{dt} \vec{\lambda}_{s_ \alpha\beta}^{\alpha} \quad (2.15)$$

di mana:

$$\vec{V}_{s_ \alpha\beta}^{\alpha} = v_{s\alpha} + jv_{s\beta}$$

Dengan operator (j), maka:

$$\begin{aligned} \vec{V}_{s_ \alpha\beta}^{\alpha} &= \vec{V}_{s_ dq} \cdot e^{j\theta} da \\ \vec{i}_{s_ \alpha\beta}^{\alpha} &= \vec{i}_{s_ dq} \cdot e^{j\theta} da \\ \vec{\lambda}_{s_ \alpha\beta}^{\alpha} &= \vec{\lambda}_{s_ dq} \cdot e^{j\theta} da \end{aligned} \quad (2.16)$$

di mana:

$$\vec{V}_{s_ dq} = V_{ds} + jV_{qs}$$

maka:

$$\vec{V}_{s_ dq}^{\alpha} \cdot e^{j\theta a} = R_s \cdot \vec{i}_{s_ dq} \cdot e^{j\theta} d + \frac{d}{dt} (\vec{\lambda}_{s_ dq} \cdot e^{j\theta da})$$

Atau

$$\vec{V}_{s_ dq}^{\alpha} \cdot e^{j\theta a} = R_s \cdot \vec{i}_{s_ dq} \cdot e^{j\theta} d + \frac{d\vec{\lambda}_{s_ dq}}{dt} \cdot e^{j\theta} da + j \frac{d\theta da}{dt} \cdot \vec{\lambda}_{s_ dq} \cdot e^{j\theta da} \quad (2.17)$$

sehingga:

$$\vec{V}_{s_dq} = R_s \cdot \vec{i}_{s_dq} + \frac{d}{dt} \cdot \vec{\lambda}_{s_dq} + j\omega_e \cdot \vec{\lambda}_{s_dq} \quad (2.18)$$

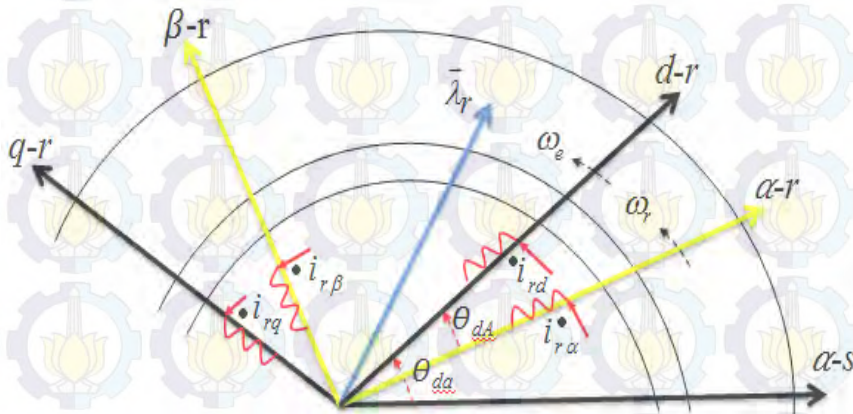
Dengan memisahkan real dan imajiner maka didapatkan tegangan stator d-q:

$$\begin{aligned} V_{ds} &= R_s \cdot i_{ds} + \frac{d}{dt} \cdot \lambda_{ds} - \omega_e \cdot \lambda_{qs} \\ V_{qs} &= R_s \cdot i_{qs} + \frac{d}{dt} \cdot \lambda_{qs} - \omega_e \cdot \lambda_{ds} \end{aligned} \quad (2.19)$$

Kemudian untuk mencari persamaan tegangan d-q, kumparan rotor dapat ditunjukkan dengan persamaan (2.20) berikut ini:

$$\begin{aligned} V_{\alpha r} &= R_s \cdot i_{\alpha r} + \frac{d}{dt} \lambda_{\alpha r} \\ V_{\beta r} &= R_s \cdot i_{\beta r} + \frac{d}{dt} \lambda_{\beta r} \end{aligned} \quad (2.20)$$

Atau dapat dilihat seperti Gambar 2.9. berikut ini:



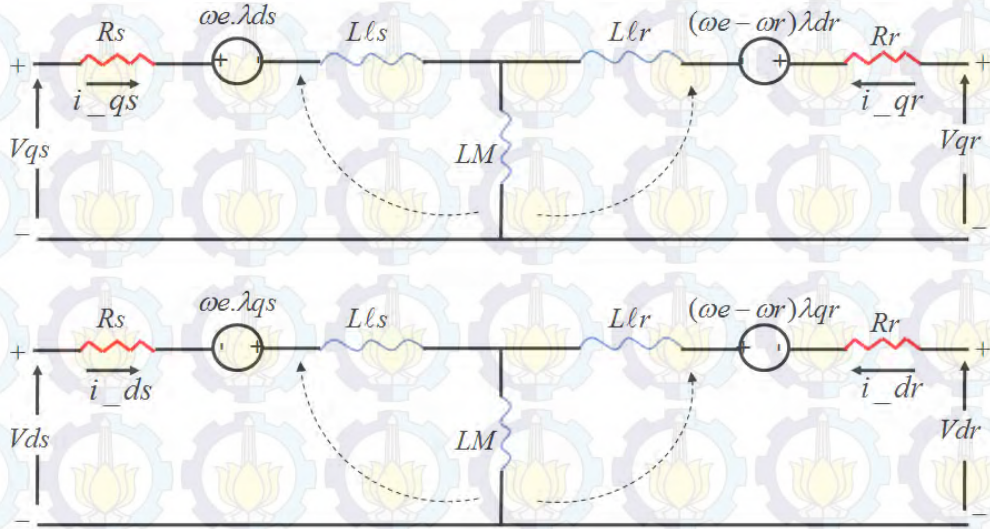
Gambar 2.9 Rangkaian Ekuivalen rotor $\alpha\beta$ dan d-q

Dengan mengikuti cara mencari tegangan pada kumparan stator, maka didapatkan tegangan rotor d-q:

$$V_{dr} = R_s \cdot i_{dr} + \frac{d}{dt} \cdot \lambda_{dr} - \omega_{dA} \cdot \lambda_{qr}$$

$$V_{qr} = R_s \cdot i_{qr} + \frac{d}{dt} \cdot \lambda_{qr} - \omega_{dA} \cdot \lambda_{dr} \quad (2.21)$$

sehingga rangkaian ekivalen motor induksi 3 fasa dalam sumbu d-q dapat dilihat pada Gambar 2.10.



Gambar 2.10 Rangkaian Ekivalen Motor Induksi Tiga Fasa Sumbu d-q

Berdasarkan rangkaian di atas maka dapat dituliskan persamaan tegangan pada stator dan rotor [11]

$$\begin{aligned} V_{qs} &= R_s \cdot i_{qs} + \omega \cdot \lambda_{ds} + p \cdot \lambda_{qs} \\ V_{ds} &= R_s \cdot i_{ds} - \omega \cdot \lambda_{qs} + p \cdot \lambda_{ds} \\ V_{qr} &= R_r \cdot i_{qr} + (\omega - \omega_r) \cdot \lambda_{dr} + p \cdot \lambda_{qr} \\ V_{dr} &= R_r \cdot i_{dr} + (\omega - \omega_r) \cdot \lambda_{qr} + p \cdot \lambda_{dr} \end{aligned} \quad (2.22)$$

di mana $p = \frac{d}{dt}$. Persamaan untuk fluks pada masing-masing kumparan dapat dinyatakan [11].

$$\begin{aligned} \lambda_{qs} &= L_s \cdot i_{qs} + L_m \cdot i_{qr} \\ \lambda_{ds} &= L_s \cdot i_{ds} + L_m \cdot i_{dr} \end{aligned}$$

$$\lambda_{qr} = L_r \cdot i_{qr} + L_m \cdot i_{qs} \quad (2.23)$$

$$\lambda_{dr} = L_r \cdot i_{dr} + L_m \cdot i_{ds}$$

sehingga apabila Persamaan (2.22) disubstitusikan ke persamaan (2.23) maka dalam bentuk matriks dapat dinyatakan menjadi:

$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \begin{bmatrix} R_s + pL_s & \omega_e \cdot L_s & pL_m & \omega_e \cdot L_m \\ -\omega_e \cdot L_s & R_s + pL_s & -\omega_e \cdot L_m & pL_m \\ pL_m & (\omega_e - \omega_r)L_m & R_r + pL_r & (\omega_e - \omega_r)L_r \\ -(\omega_e - \omega_r)L_m & pL_m & -(\omega_e - \omega_r)L_r & R_r + pL_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad (2.24)$$

dengan,

$$L_s = L_{ls} + L_m$$

$$L_r = L_{lr} + L_m \quad (2.25)$$

di mana,

V_{qs}, V_{ds} = tegangan stator pada sumbu dq (Volt)

V_{qr}, V_{dr} = tegangan rotor pada sumbu dq (Volt)

i_{qs}, i_{ds} = arus stator pada sumbu dq (Ampere)

i_{qr}, i_{dr} = arus rotor pada sumbu dq (Ampere)

$\lambda_{qs}, \lambda_{ds}$ = fluks stator pada sumbu dq (Webber)

$\lambda_{qr}, \lambda_{dr}$ = fluks rotor pada sumbu dq (Webber)

R_s = tahanan stator (Ohm)

R_r = tahanan rotor (Ohm)

L_s = induktansi stator (Henry)

L_r = induktansi diri rotor (Henry)

L_M = induktansi mutual (Henry).

Dari nilai arus dan fluks yang sudah dipaparkan pada persamaan di atas, maka dapat dicari nilai torsi elektromagnetik yang dibangkitkan dengan menggunakan persamaan (2.27)

$$T_{em} = \frac{3}{2} \frac{P}{2} (\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds}) \quad (2.26)$$

di mana,

T_{em} = Torsi elektromagnetik yang dibangkitkan (Nm)

P = Jumlah pasang kutub.

Untuk mencari besarnya kecepatan yang dihasilkan pada rotor (ω_r), dapat menggunakan persamaan (2.27) di mana pada persamaan tersebut terdapat hubungan antara torsi yang dibangkitkan (T_{em}) dengan torsi beban (T_L)

$$T_{em} = T_L + J \frac{d\omega_m}{dt} = T_L + \frac{2}{P} J \frac{d\omega_r}{dt}$$

$$\frac{d}{dt} \omega_m = \frac{T_{em} - T_L}{J_{eq}} \quad (2.27)$$

di mana,

T_L = torsi beban (N.m)

T_{em} = torsi elektromagnetik yang dibangkitkan (N.m)

J = momen inersia (kg.m²)

P = jumlah pasang kutub

ω_r = kecepatan elektrik rotor (rad/s)

ω_m = kecepatan mekanik rotor (rad/s)

Persamaan (2.24) dilakukan penyederhanaan agar memudahkan dalam pembuatan simulasi dalam *software* yang dipakai. Proses penyederhanaan persamaan (2.24) dapat dilihat di bawah ini:

$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \begin{bmatrix} R_s + pL_s & \omega_r L_s & pL_m & \omega_r L_m \\ -\omega_r L_s & R_s + pL_s & -\omega_r L_m & pL_m \\ pL_m & (\omega_r - \omega_r)L_m & R_r + pL_r & (\omega_r - \omega_r)L_r \\ -(\omega_r - \omega_r)L_m & pL_m & -(\omega_r - \omega_r)L_r & R_r + pL_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix}$$

dilakukan pemisahan antara variabel yang mengandung turunan, sehingga bentuk persamaannya menjadi:

$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \begin{bmatrix} R_s & \omega \cdot L_s & 0 & \omega \cdot L_m \\ -\omega \cdot L_s & R_s & -\omega \cdot L_m & 0 \\ 0 & (\omega - \omega_r) L_m & R_r & (\omega - \omega_r) L_r \\ -(\omega - \omega_r) L_m & 0 & -(\omega - \omega_r) L_r & R_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \\ + \begin{bmatrix} pL_s & 0 & pL_m & 0 \\ 0 & pL_s & 0 & pL_m \\ pL_m & 0 & pL_r & 0 \\ 0 & pL_m & 0 & pL_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix}$$

dalam bentuk lain, persamaan di atas dapat menjadi seperti dipersamaan berikut ini:

$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \begin{bmatrix} R_s & \omega \cdot L_s & 0 & \omega \cdot L_m \\ -\omega \cdot L_s & R_s & -\omega \cdot L_m & 0 \\ 0 & (\omega - \omega_r) L_m & R_r & (\omega - \omega_r) L_r \\ -(\omega - \omega_r) L_m & 0 & -(\omega - \omega_r) L_r & R_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \\ + \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & pL_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix}$$

misal, $\mathbf{P} = \begin{bmatrix} R_s & \omega \cdot L_s & 0 & \omega \cdot L_m \\ -\omega \cdot L_s & R_s & -\omega \cdot L_m & 0 \\ 0 & (\omega - \omega_r) L_m & R_r & (\omega - \omega_r) L_r \\ -(\omega - \omega_r) L_m & 0 & -(\omega - \omega_r) L_r & R_r \end{bmatrix}$

dan $\mathbf{Q} = \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & pL_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix}$. Persamaan di atas dapat dibentuk menjadi lebih

sederhana yaitu:

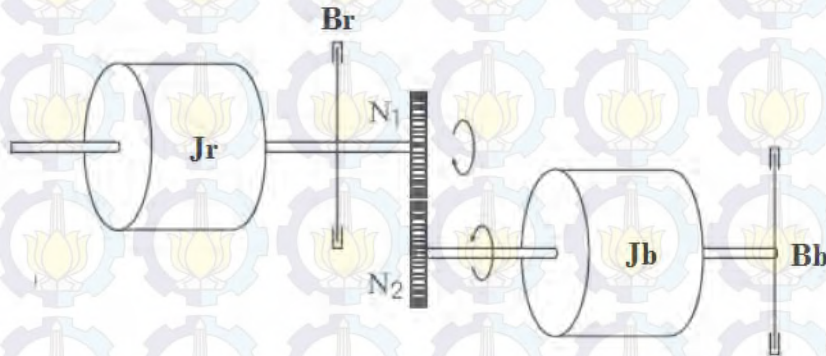
$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \mathbf{P} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} + \mathbf{Q} \frac{d}{dt} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad (2.28)$$

Apabila yang ingin dicari nilai arus, maka persamaan di atas menjadi:

$$\frac{d}{dt} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} = \mathbf{Q}^{-1} \begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} - \mathbf{Q}^{-1} \mathbf{P} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad (2.29)$$

2.2.4 Model Dinamik Beban Motor Induksi

Torsi beban(T_L) untuk model motor induksi menggunakan pembebanan pada poros beban, dapat dilihat pada Gambar 2.11:



Gambar 2.11 Model Dinamik Beban Inersia Dan Damper

Dengan persamaan matematika untuk model dinamik beban inersia dan damper sebagai berikut:

$$\tau_e - \left(\frac{N_1}{N_2}\right)^2 \left(B_b \omega_m + J_b \frac{d}{dt} \omega_m \right) - B_r \omega_m = J_r \frac{d}{dt} \omega_m \quad (2.30)$$

di mana:

T_e = Torsi elektromagnetik yang dibangkitkan (Nm),

J_r = Momen inersia motor (kg.m^2),

J_b = Momen inersia beban (kg.m^2),

N_1 = Jumlah gigi pada roda gigi 1,

N_2 = Jumlah gigi pada roda gigi 2,

B_m = Damper motor,

B_b = Damper beban,

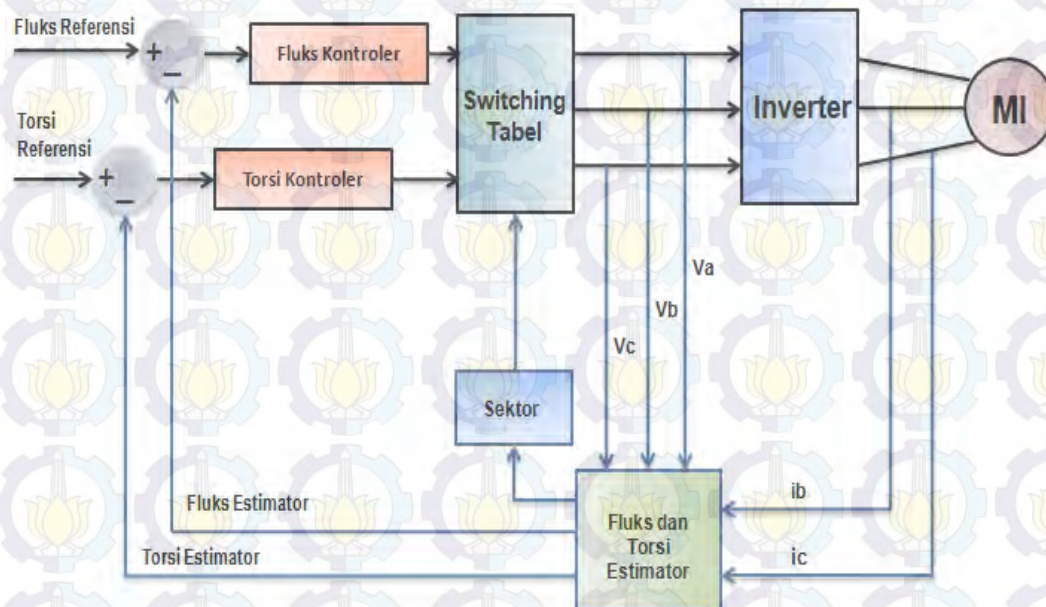
ω_m = Kecepatan mekanik rotor (rad/s).

2.3 Direct Torque Control (DTC)

Direct Torque Control pertama kali dikembangkan oleh Takahasi dan Noguchi tahun 1986. Dasar dari metode DTC adalah perubahan torsi sebanding dengan slip antara fluks stator dan fluks rotor pada kondisi fluks bocor stator tetap[3]. Pada motor induksi dengan tipe rotor sangkar untuk waktu tetap rotor menjadi sangat besar, fluks bocor berubah perlahan dibanding dengan perubahan fluks bocor stator. Oleh karena itu pada keadaan perubahan cepat fluks rotor cenderung tidak berubah. Perubahan cepat dari torsi elektromagnetik dapat dihasilkan dari putaran fluks stator, sebagai arah torsi.

Dengan kata lain fluks stator dapat seketika mempercepat atau memperlambat dengan menggunakan vektor tegangan stator yang sesuai. Torsi dan fluks kontrol bersama-sama dan decouple dicapai dengan pengaturan langsung dari tegangan stator, dari error respon torsi dan fluks.

DTC biasanya digunakan sesuai vektor tegangan dalam hal ini untuk memelihara torsi dan fluks stator dengan dua daerah histerisis [3]. Rangkaian DTC dapat ditunjukkan pada Gambar 2.12.



Gambar 2.12 Diagram Blok Sistem DTC Motor Induksi Tiga Fasa

Gambar 2.11 menunjukkan bahwa *Direct Torque Control* (DTC) merupakan salah satu strategi vektor kontrol berdasarkan pengontrolan fluks dan torsi

elektromagnetik secara langsung pada motor induksi dengan penentuan pemilihan vektor tegangan.

Dari pengertian mengenai DTC disebutkan bahwa DTC merupakan suatu strategi vektor kontrol. Strategi terdiri dua loop kontrol yang berhubungan dengan besarnya fluks stator dan torsi elektromagnetik. Masing-masing loop kontrol mempunyai kontroler/pembanding histerisis yang menunjukkan aksi kontrol yang harus diambil (untuk menambah atau mengurangi besarnya fluks stator dan torsi). Tabel *switching* disusun untuk memberikan posisi vektor fluks pada saat ini dan aksi kontrol yang diinginkan. Vektor tegangan yang tepat untuk diaplikasikan pada motor dapat dipilih.

Beberapa kelebihan penggunaan kontrol DTC adalah sebagai berikut [6]:

- a. Performansi bagus dan bisa dilakukan tanpa menggunakan sensor kecepatan.
- b. Tidak ada umpan balik kontrol arus.
- c. Tidak ada algoritma *Pulse Width Modulation* (PWM) yang diterapkan.
- d. Tidak ada transformasi vektor seperti pada vektor kontrol.
- e. Pemrosesan sinyal umpan balik hampir sama dengan fluks stator pada kontrol vektor.
- f. Kontrol batas histerisis menghasilkan riak (ripple) fluks dan torsi dan frekuensi *switching* yang tidak konstan (seperti batas histerisis kontrol arus).

2.3.1 Prinsip Dasar DTC

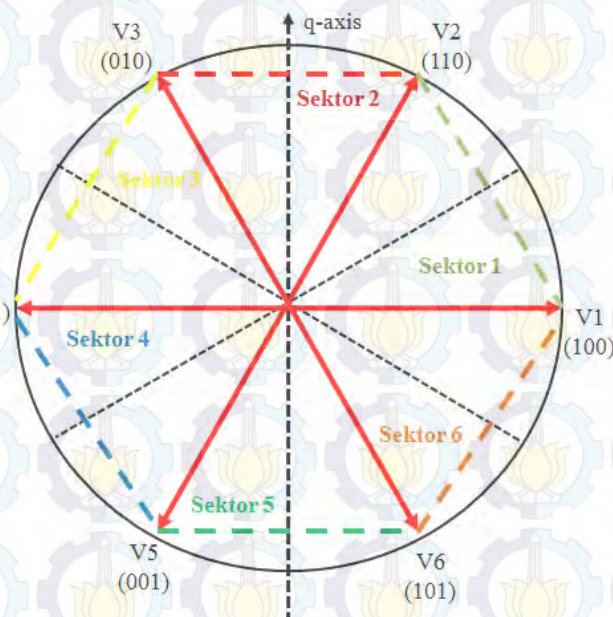
Prinsip dan pemikiran dasar dari metode DTC yang diusulkan oleh Takahashi dan Noguchi pada tahun 1986, dapat diformulasikan sebagai berikut [4]:

- a. Fluks stator merupakan perubahan atau penambahan nilai tegangan EMF. Besarnya nilai fluks stator sangat bergantung pada tegangan stator.
- b. Torsi elektromagnetik yang dibangkitkan berbanding lurus dengan nilai sinusoidal dari sudut antara vektor fluks stator dan rotor.
- c. Perubahan nilai fluks rotor terhadap tegangan stator lebih lambat dibandingkan fluks stator.

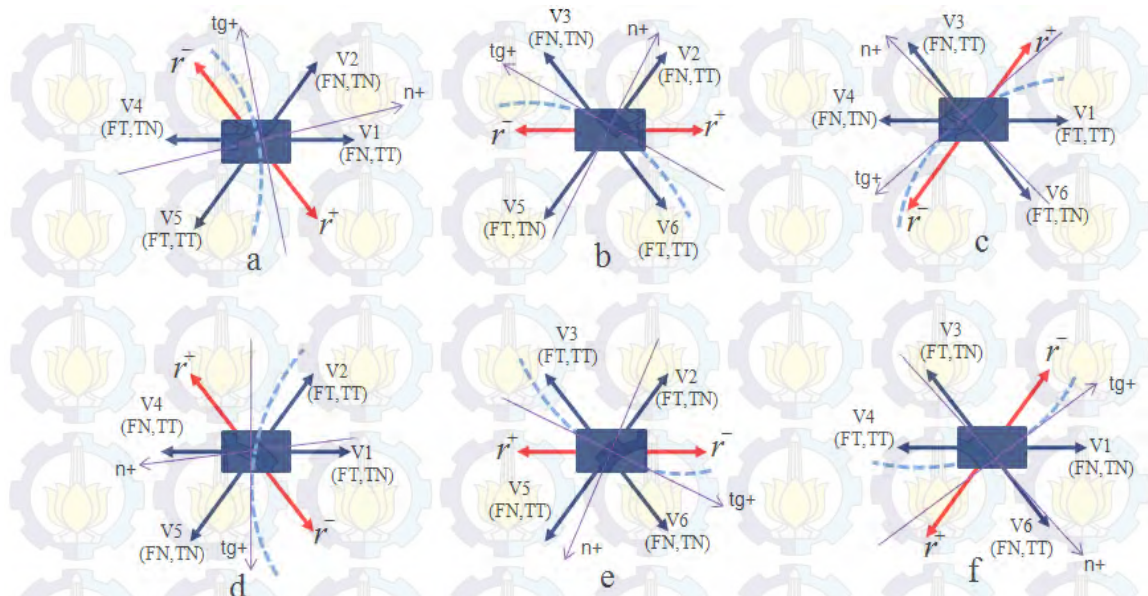
Berdasarkan prinsip dasar DTC yang dijelaskan di atas, besarnya nilai fluks stator dan torsi elektromagnetik yang dibangkitkan dapat secara langsung dikontrol dengan memilih secara tepat vektor tegangan stator, yaitu pemilihan *state* inverter secara berurutan dengan spesifikasi berikut:

- Vektor tegangan bukan nol yang tidak sejajar dengan vektor fluks stator tidak melebihi $\pm 90^\circ$ menyebabkan fluks naik.
- Vektor tegangan bukan nol yang tidak sejajar dengan vektor fluks stator melebihi $\pm 90^\circ$ menyebabkan fluks turun.
- State* nol, V_0 dan V_7 (pada saat durasi pendek) tidak berpengaruh pada vektor fluks stator yang berarti motor berhenti bergerak.
- Torsi elektromagnetik yang dibangkitkan dapat dikontrol dengan memilih *state* inverter yang vektor fluks statornya dipercepat, diberhentikan atau diperlambat.

Bidang dq dibagi menjadi enam sektor yang masing-masing sektor besarnya 60° (mulai dari sektor 1 sampai sektor enam). Pembagian keenam sektor dari bidang dq dapat dilihat pada Gambar 2.13. Sebuah vektor fluks stator dikatakan berhubungan dengan vektor tegangan ketika melewati sektor yang sama.



Gambar 2.13 Pembagian Ruang Vektor Tegangan Bidang d-q



Gambar 2.14 Vektor Tegangan Saat Fluks Stator Dalam a) sektor 1, b) sektor 2, c) sektor 3, d) sektor 4, e) sektor 5, f) sektor 6

Pengaruh dari tiap-tiap vektor tegangan terhadap fluks stator dan torsi elektromagnetik dapat dilihat pada Tabel 2.1. Pengaruh dari vektor tegangan V_1 dan V_4 pada nilai torsi elektromagnetik adalah ambigu karena tergantung pada apakah vektor fluks mendahului atau tertinggal dari vektor tegangan. Vektor tegangan yang bernilai nol (V_0 dan V_7) tidak berpengaruh pada fluks stator akan tetapi menurunkan nilai torsi elektromagnetik.

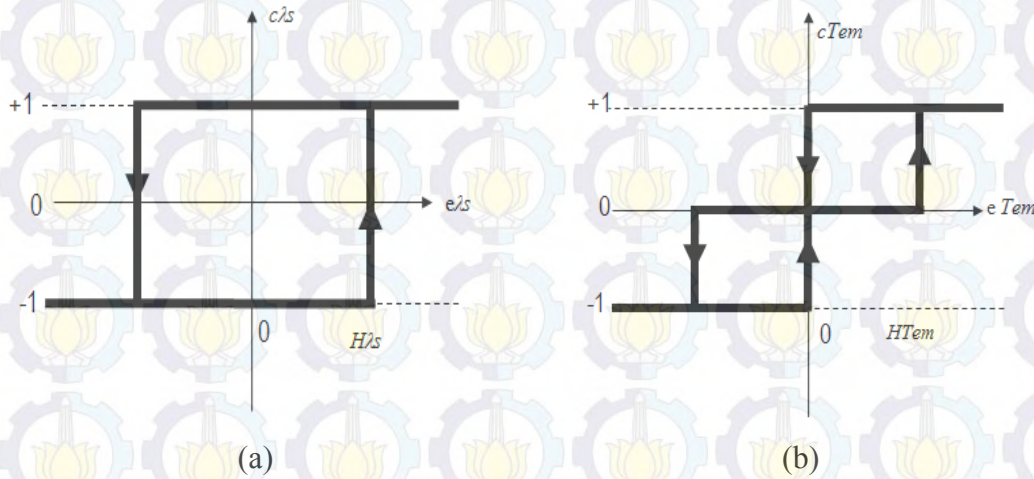
Tabel 2.1 Pengaruh Tiap-Tiap Vektor Tegangan terhadap Nilai Fluks Stator dan Torsi Elektromagnetik

| | V_1 | V_2 | V_3 | V_4 | V_5 | V_6 | V_0/V_7 |
|-------------|------------|------------|------------|------------|------------|------------|------------|
| λ_s | \nearrow | \nearrow | \searrow | \searrow | \searrow | \nearrow | - |
| T_e | ? | \nearrow | \nearrow | ? | \searrow | \searrow | \searrow |

2.3.2 Kontroler Histerisis

Kontroler digunakan untuk mendapatkan nilai kesalahan fluks dan torsi pada daerah histeresisnya. Pada kesalahan fluks dihasilkan dua nilai histerisis (*two-level hysteresis*), yaitu -1 atau 1 sedangkan pada kesalahan torsi dihasilkan tiga nilai histerisis (*three-level hysteresis*) yaitu -1, 0, 1. Nilai kesalahan (*error*)

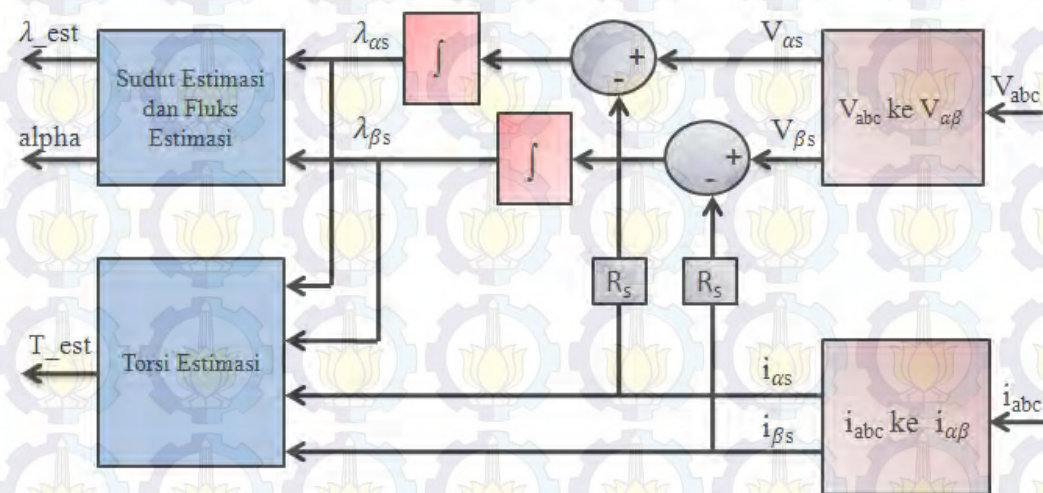
fluks dan torsi didapatkan dari operasi pengurangan antara nilai referensi dengan nilai estimasi. Kontroler histerisis fluks dan torsi elektromagnetik dapat dilihat pada Gambar 2.15.



Gambar 2.15 (a) Kontroler Histerisis Fluks, (b) Kontroler Histerisis Torsi

2.3.3 Estimator dan Sektor Fluks Stator

Pada sistem DTC, estimator berfungsi sebagai pengganti sensor pada umpan balik sistem. Pada estimator ini akan dihitung besarnya nilai fluks dan torsi sebagai umpan balik untuk masukan fluks dan torsi referensi. Selain dihitung besarnya nilai fluks dan torsi, pada estimator juga dihitung besarnya sudut dari fluks stator yang dipergunakan untuk masukan dalam menentukan sektor fluks stator.



Gambar 2.16 Diagram Blok Estimator *Direct Torque Control* (DTC)

Pada Gambar 2.16. Diagram blok di atas, untuk mengubah sistem koordinat stasioner tiga fasa (a,b,c) menjadi sistem koordinat dua fasa ($\alpha\beta$) yang stasioner atau disebut dengan transformasi clarke yaitu yang terdapat pada blok i_{abc} ke $i_{\alpha\beta}$ dan V_{abc} ke $V_{\alpha\beta}$ dihitung dengan menggunakan persamaan (2.8) yaitu:

$$\begin{bmatrix} X_{\alpha s} \\ X_{\beta s} \end{bmatrix} = k \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2}\sqrt{3} & -\frac{1}{2}\sqrt{3} \end{bmatrix} \begin{bmatrix} X_{as} \\ X_{bs} \\ X_{cs} \end{bmatrix}$$

Dengan k pada transformasi clarke $= \frac{2}{3}$

sehingga $\lambda_{\alpha\beta}$:

$$\lambda_{\alpha s} = (V_{\alpha s} - R_s \cdot i_{\alpha s}) \quad (2.31)$$

$$\lambda_{\beta s} = (V_{\beta s} - R_s \cdot i_{\beta s}) \quad (2.32)$$

sedangkan untuk nilai magnitud dan sudut fluks stator estimasi dihitung dengan menggunakan persamaan (2.33) dan (2.34).

$$|\lambda_s| = \sqrt{\lambda_{ds}^2 + \lambda_{qs}^2} \quad (2.33)$$

$$\alpha = \angle \lambda_s = \tan^{-1} \frac{\lambda_{qs}}{\lambda_{ds}} \quad (2.34)$$

untuk torsi estimasi didapatkan dari persamaan (2.26) yaitu:

$$T_{em} = \frac{3P}{2} (\lambda_{\alpha s} \cdot i_{\beta s} - \lambda_{\beta s} \cdot i_{\alpha s})$$

2.3.4 Tabel *Switching*

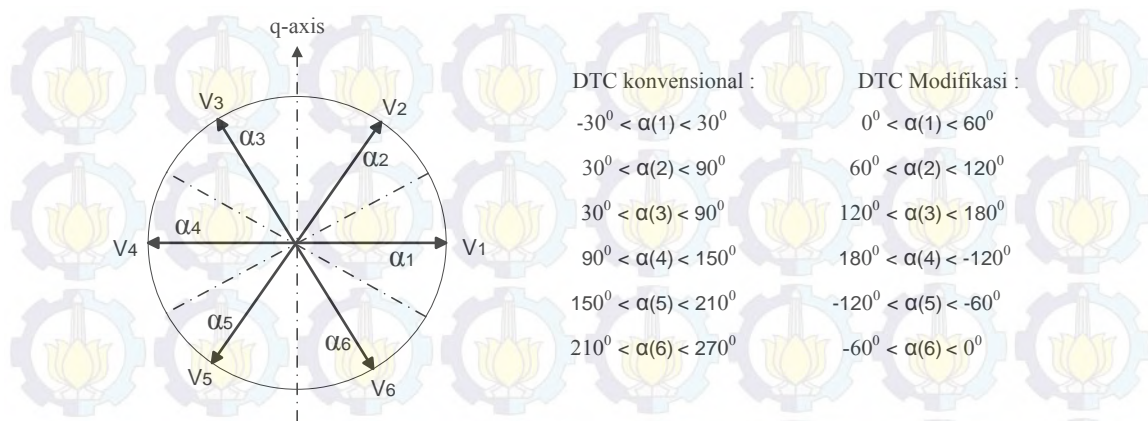
Sinyal masukan untuk tabel *switching* diperoleh dari nilai output dari fluks dan torsi histerisis serta nilai dari sektor fluks stator (α). Algoritma DTC memilih vektor tegangan inverter untuk diterapkan pada motor induksi sebagai sumber

tegangan. Vektor tegangan aktif atau bernilai bukan nol terdiri dari \bar{V}_1 (100); \bar{V}_2 (110); \bar{V}_3 (010); \bar{V}_4 (011); \bar{V}_5 (001); \bar{V}_6 (101). Sedangkan vektor tegangan bernilai nol adalah \bar{V}_0 (000); \bar{V}_7 (111). Aturan pemilihan tegangan pada tabel *switching* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Tabel *Switching* dari Vektor Tegangan Inverter [10]

| H_ψ | HT_e | Sektor 1 $\alpha(1)$ | Sektor 2 $\alpha(2)$ | Sektor 3 $\alpha(3)$ | Sektor 4 $\alpha(4)$ | Sektor 5 $\alpha(5)$ | Sektor 6 $\alpha(6)$ |
|----------|--------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| 1 | 1 | \bar{V}_2 | \bar{V}_3 | \bar{V}_4 | \bar{V}_5 | \bar{V}_6 | \bar{V}_1 |
| | 0 | \bar{V}_0 | \bar{V}_7 | \bar{V}_0 | \bar{V}_7 | \bar{V}_0 | \bar{V}_7 |
| | -1 | \bar{V}_6 | \bar{V}_1 | \bar{V}_2 | \bar{V}_3 | \bar{V}_4 | \bar{V}_5 |
| -1 | 1 | \bar{V}_3 | \bar{V}_4 | \bar{V}_5 | \bar{V}_6 | \bar{V}_1 | \bar{V}_2 |
| | 0 | \bar{V}_7 | \bar{V}_0 | \bar{V}_7 | \bar{V}_0 | \bar{V}_7 | \bar{V}_0 |
| | -1 | \bar{V}_5 | \bar{V}_6 | \bar{V}_1 | \bar{V}_2 | \bar{V}_3 | \bar{V}_4 |

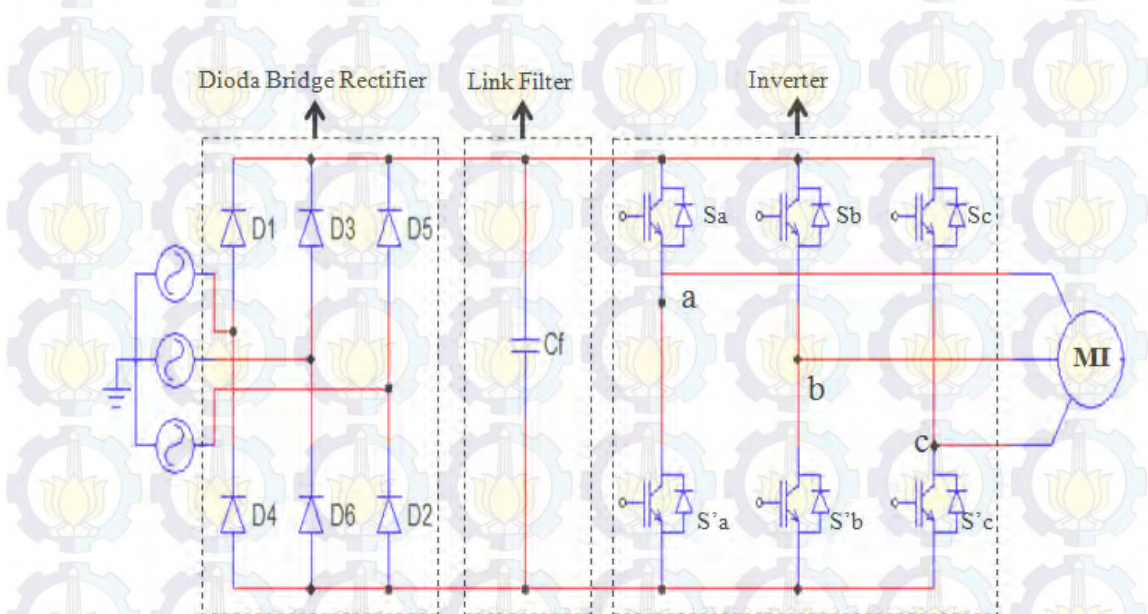
Pada DTC konvensional, *state* inverter pada V_1 dan V_4 tidak digunakan karena nilai torsi tidak jelas apakah naik atau turun, tergantung pada posisi sektornya apakah berada pada posisi 30° pertama atau kedua. Sedangkan pada DTC modifikasi, *state* inverter yang tidak digunakan adalah V_3 dan V_6 karena nilai fluks stator pada kondisi *state* ini ambigu. Hal ini dianggap menjadi suatu keuntungan dalam mengontrol torsi dengan menggunakan metode DTC modifikasi. Lebih baik kehilangan penggunaan dua nilai fluks yang ambigu dibandingkan torsi. Perbedaan pembagian sektor antara DTC Konvensional dengan DTC Modifikasi dapat dilihat pada Gambar 2.17 [10].



Gambar 2.17 Vektor Tegangan Inverter dan Sektor *Switching* Fluks Stator pada DTC Konvensional dan DTC Modifikasi

2.3.5 Inverter Sumber Tegangan

Inverter Sumber Tegangan memiliki enam buah saklar (tiga pasang) yang bekerja secara bergantian. Inverter akan menghasilkan sumber tegangan tiga fasa sebagai masukan motor induksi. Tiap pasangannya terdapat dua buah saklar yang bekerja secara bergantian. Hal ini berarti tidak ada kemungkinan dari satu pasang saklar bekerja bersamaan karena dapat mengakibatkan terjadinya hubung singkat. Rangkaian inverter tiga fasa ditunjukkan pada Gambar 2.18.



Gambar 2.18 Rangkaian Inverter Tiga Fasa

Pada inverter tiga fasa (Gambar 2.18) dapat dibuat beberapa kombinasi switching yang akan menghasilkan vector tegangan yang berbeda. Jika S_a *on* dan

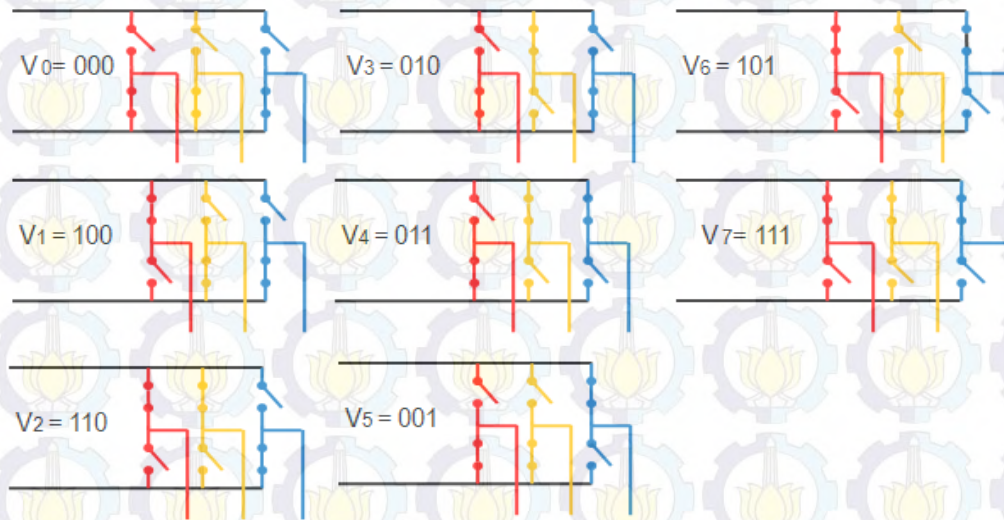
S'a *off* (dilambangkan dengan Sa=1) maka pada V_a timbul tegangan sebesar V_{dc} . Demikain juga halnya pada Sb dan Sc. Vektor tegangan yang terjadi untuk tegangan *line to line* berlaku :

$$\begin{cases} v_{ab} = v_a - v_b \\ v_{bc} = v_b - v_c \\ v_{ca} = v_c - v_a \end{cases} \quad (2.35)$$

sehingga output tegangan inverter dapat ditentukan dengan menggunakan persamaan matriks berikut:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \frac{1}{3} V_{dc} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (2.36)$$

Urutan *switching* pada Inverter tiga fasa dapat dilihat seperti pada Gambar 2.19.



Gambar 2.19 Urutan *Switching* Inverter Tiga Fasa

Dari 8 jenis urutan *switching* yang berbeda seperti pada Gambar 2.19 terdapat dua vektor tegangan pasif yaitu V_0 yang nilainya '000' dan V_7 yang nilainya '111' dan enam vektor tegangan aktif yaitu $V_1, V_2, V_3, V_4, V_5, V_6$. Dari 8 urutan *switching* ini dapat dibuat dari kombinasi Sa, Sb dan Sc seperti kita lihat pada tabel 2.3.

Tabel 2.3 Kondisi *Switching Inverter*

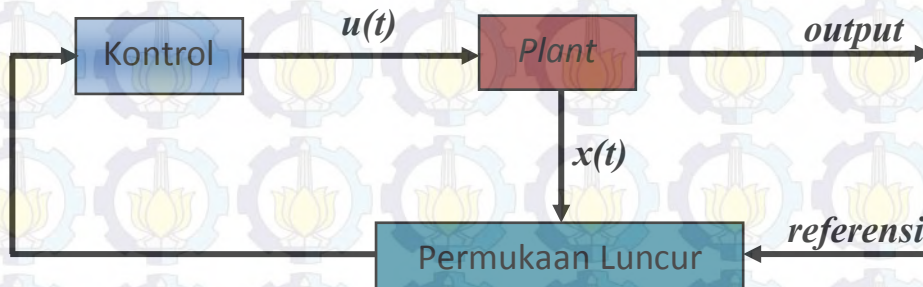
| Sektor | Switch (on) | V _a | V _b | V _c | Vektor Tegangan |
|--------|--|---------------------|---------------------|---------------------|----------------------|
| 0 | S'a,S'b,S'c | 0 | 0 | 0 | V ₀ (000) |
| 1 | S _a ,S'b,S'c | 2/3V _{dc} | -1/3V _{dc} | -1/3V _{dc} | V ₁ (100) |
| 2 | S _a ,S _b ,S'c | 1/3V _{dc} | 1/3V _{dc} | -2/3V _{dc} | V ₂ (110) |
| 3 | S'a,S _b ,S'c | -1/3V _{dc} | 2/3V _{dc} | -1/3V _{dc} | V ₃ (010) |
| 4 | S'a,S _b ,S _c | -2/3V _{dc} | 1/3V _{dc} | 1/3V _{dc} | V ₄ (011) |
| 5 | S'a,S'b,S _c | -1/3V _{dc} | -1/3V _{dc} | 2/3V _{dc} | V ₅ (001) |
| 6 | S _a ,S'b,S _c | 1/3V _{dc} | -2/3V _{dc} | 1/3V _{dc} | V ₆ (101) |
| 7 | S _a ,S _b ,S _c | 0 | 0 | 0 | V ₇ (111) |

2.4 *Sliding Mode Control* (SMC)[13]

Dalam merumuskan berbagai persoalan kontrol, biasanya akan ditemui ketidaksesuaian antara *plant* yang sebenarnya dengan model yang dibangun untuk mendesain pengaturan. Ketidaksesuaian ini bisa disebabkan karena terjadinya perubahan parameter sistem atau karna penyederhanaan yang diambil untuk memodelkan kelakuan sistem yang kompleks. Karena itu di dalam mendesain pengaturan haruslah dipertimbangkan faktor adanya ketidaksesuaian tersebut diatas, sehingga kinerja sistem kontrol yang diharapkan dapat tercapai. Penelitian tentang hal ini telah memicu berkembangnya teori kontrol kokoh (*robust control*) dan kontrol adaptif (*adaptive control*). Metode kontrol modus luncur (*Sliding Mode Control*, *SMC*) merupakan salah satu metode kontrol yang memiliki sifat kekokohan [5].

Ide awal metode modus luncur dikemukakan oleh Itkis dan Utkin. Metode ini dikembangkan dengan maksud untuk memecahkan masalah dalam sistem kontrol yang timbul akibat adanya ketidakakuratan model sistem. Pengguna metode modus luncur, terutama pada sistem kontrol non linier, pada dasarnya merupakan pendekatan yang sederhana dari sistem kontrol kokoh (*robust control*).

Sliding Mode Control pada intinya adalah memilih suatu masukan sinyal kontrol (*control input, $u(t)$*) tertentu untuk sistem yang akan dikontrol, terutama sistem non linier, yang akan membawa dinamika sistem masuk kedalam suatu permukaan luncur (*sliding surface*), dan selanjutnya akan menyebabkan status sistem meluncur (*sliding*) ke titik seimbang. Secara umum sistem kontrol menggunakan metode modus luncur dapat diilustrasikan seperti pada gambar 2.20.



Gambar 2.20 Diagram kontrol modus luncur

Sliding Mode Control merupakan sebuah kontrol umpan balik pensakralan berkecepatan tinggi yang efektif dan kokoh dalam mengontrol sistem linier dan sistem non linier. Sistem kontrol ini kokoh karena menyediakan sebuah metode perancangan sistem yang tidak peka terhadap ketidakpastian parameter lingkungan dan gangguan luar.

Pada prinsipnya *Sliding Mode Control* menggunakan sebuah hukum kontrol pensakralan berkecepatan tinggi untuk membawa *trajektori* status dari sistem linier atau sistem non linier ke dalam sebuah permukaan (*hyperplane*) tertentu dalam ruang status (disebut permukaan luncur / *sliding surface*), kemudian *trajektori* status tersebut dipelihara agar tetap meluncur pada permukaan tersebut. Proser pemeliharaan *trajektori* status pada permukaan luncur mengakibatkan terjadinya osilasi pada permukaan luncur. Osilasi ini sering disebut *chattering*. Fenomena *chattering* pada permukaan luncur akan berdampak pada stabilitas dari sistem kontrol.

Pada *Sliding Mode Control*, waktu yang dibutuhkan oleh status dalam mencapai permukaan luncur akan berdampak terhadap kecepatan tanggapan sistem terhadap waktu. Waktu yang dibutuhkan oleh status dalam mencapai

permukaan luncur sering disebut dengan *Hitting Time*. Salah satu keuntungan dari *Sliding Mode Control* adalah ketika sistem mengenai permukaan luncur, maka sistem tersebut tidak akan peka terhadap ketidakpastian parameter lingkungan dan gangguan luar. Untuk menjaga perilaku sistem sehingga tidak peka terhadap perubahan lingkungan, maka dibutuhkan *Hitting Time* yang kecil agar sistem cepat mengenai permukaan luncur. *Hitting Time* dan *Chattering* yang kecil dua hal yang sangat penting dalam merancang sistem *Sliding Mode Control*. Sebuah sistem linier SISO direpresentasikan dengan persamaan (2.37) dan (2.38) yang merupakan persamaan dari sistem:

$$\dot{x} = Ax + Bu \quad (2.37)$$

Dengan

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -a_n & -a_{n-1} & a_{n-2} & \dots & -a_2 & -a_1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{bmatrix} \quad (2.38)$$

di mana $x(t) = [x_1 \dots x_n]^T$ merupakan sebuah vektor status berdimensi-n, A merupakan sebuah matriks sistem n x n, dan B adalah sebuah vektor input berdimensi-n. Secara umum ada dua tahap dalam perancangan *Sliding Mode Control*, yaitu:

1. Perancangan Permukaan Luncur

Permukaan luncur (*Sliding Surface*) dirancang dengan membuat tanggapan sistem dengan persamaan:

$$\tau(x) = Sx(t) = 0 \quad (2.39)$$

di mana,

$$S = [S_1 \dots S_n] \quad (2.40)$$

adalah vektor koefisien *Sliding Surface*. Maka, dengan substitusi persamaan (2.39) dan (2.40) maka akan didapatkan persamaan *Sliding* seperti persamaan (2.41) berikut ini:

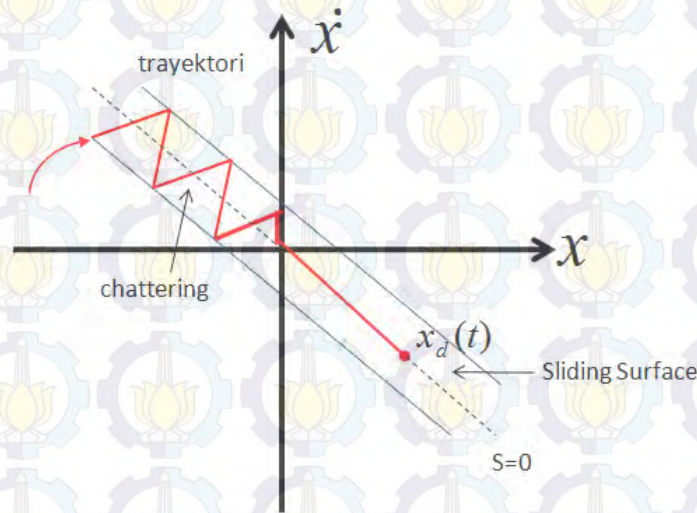
$$\tau(x) = s_1 x_1(t) + s_2 x_2(t) + \dots s_n x_n(t) = 0 \quad (2.41)$$

2. Perancangan *Sliding Mode Control* (SMC)

Pada perancangan ini hukum kontrol $u(t)$ dibuat dengan menggunakan syarat kestabilan lyapunov $\dot{\tau}^T \tau < 0$. secara umum hukum kontrol dapat dipisah menjadi dua bagian sinyal kontrol yaitu u_{eq} dan u_n , sehingga hukum kontrol sistem diperoleh dengan menjumlahkan kedua bagian sinyal kontrol tersebut, seperti yang terlihat pada persamaan (2.42)

$$u(t) = u_{eq} + u_n = -Kx \quad (2.42)$$

Dimana u_{eq} merupakan sinyal kontrol ekivalen yang akan membawa *trajektori* status kepermukaan luncur, sedangkan u_n merupakan sinyal kontrol natural untuk menjaga agar *trajektori* status tetap berada pada permukaan luncur, seperti yang terlihat pada Gambar 2.21.



Gambar 2.21 Diagram Fasa *Trajektori* Status

Sistem loop tertutup persamaan (2.43) diperoleh dari persamaan (2.38) dan persamaan (2.41) :

$$\dot{x} = Ax + B(u_{eq} + u_n) = Ax + Bu_{eq} + Bu_n \quad (2.43)$$

Sementara sinyal kontrol ekivalen persamaan (2.44) diperoleh dengan mensubstitusikan persamaan (2.39) dan persamaan (2.43), dengan syarat:

$$Bu_n = 0$$

$$\tau = Sx = 0$$

$$\tau = S\dot{x} = S(Ax + Bu_{eq}) = 0$$

$$u_{eq} = -(SB)^{-1}SAx \quad (2.44)$$

$$u_n = -k\text{sign}(\tau) \quad (2.45)$$

Dengan substitusi persamaan substitusi persamaan (2.37) dan (2.44) maka diperoleh sistem loop tertutup pada *Sliding Surface*, seperti yang terlihat pada persamaan (2.46) :

$$\dot{x} = [Ax - B(SB)^{-1}SA] \quad (2.46)$$

Persamaan (2.47) merupakan sinyal kontrol untuk mempertahankan *trajektori* status pada permukaan luncur.

$$u_n = -k(SB(x, t))^{-1} \text{sign}(\tau) \text{ jika } (SB(x, t)) \text{ invertible} \quad (2.47)$$

2.5 Algoritma Genetika [14]

Algoritma Genetika merupakan suatu metode yang menggunakan seleksi alam yang merupakan bagian utama dari prinsip evolusi sebagai dasar pemikiran untuk menyelesaikan suatu permasalahan. Prinsip ini dikemukakan oleh Charles Darwin, di mana tanpa menghiraukan prinsip dasar penurunan sifat, Darwin mengemukakan penggabungan kualitas induk pada generasi berikutnya, di samping itu bahwa individu yang mampu beradaptasi dengan lingkungannya akan mempunyai kesempatan hidup yang lebih besar. Penggunaan prinsip genetika pada komputer dimulai pada tahun 1950 ketika beberapa ahli Biologi menggunakan komputer untuk simulasi sistem biologi. Akhir tahun 1975 John Holland dari Universitas Michigan melalui paper yang berjudul “*Adaption in Natural and Artificial System*” menggunakan konsep dasar *Algoritma Genetika*. *Algoritma*

Genetika bekerja dengan suatu populasi string dan melakukan proses pencarian nilai optimal secara parallel, dengan menggunakan operator genetika. *Algoritma Genetika* akan melakukan rekombinasi antar individu. Algoritma genetika memiliki elemen dasar berupa string yang tersusun dari rangkaian substring (gen), yang masing-masing merupakan kode dari parameter dalam ruang solusi dimana suatu string (kromosom) menyatakan kandidat solusi. Kumpulan string dalam populasi berkembang dari generasi ke generasi melalui operator genetika. Pada setiap iterasi, individu-individu (Kromosom) dalam populasi itu akan dievolusi dan diseleksi untuk menentukan populasi pada generasi berikutnya. Populasi ini akan terus berulang sampai menemukan suatu parameter dengan nilai yang paling optimal sesuai dengan yang diinginkan.

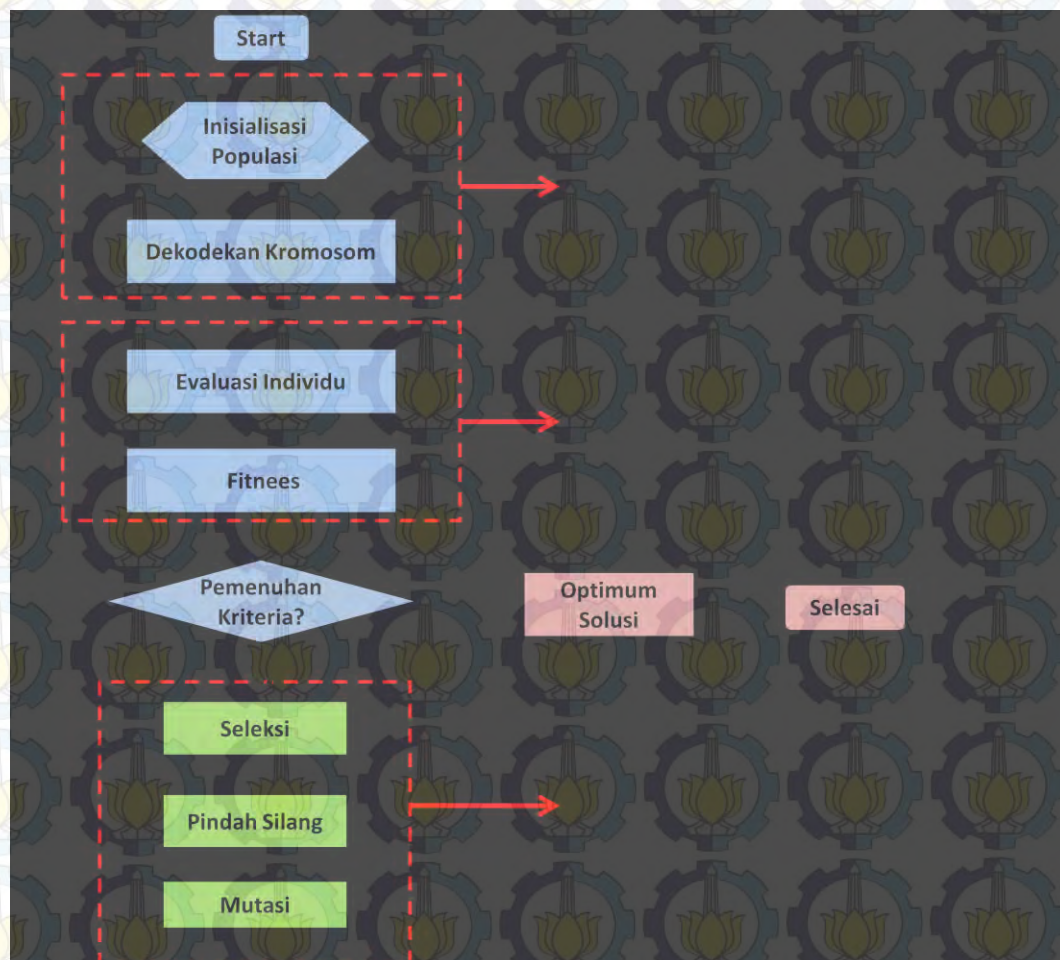
2.5.1 Aplikasi *Algoritma Genetika*

Sejak pertama kali dirintis oleh John Holland, *Algoritma Genetika* telah dipelajari, diteliti dan diaplikasikan secara luas pada berbagai bidang. Algoritma Genetika banyak digunakan pada masalah praktis yang berfokus pada pencarian parameter-parameter yang optimal. Namun demikian, *Algoritma Genetika* juga dapat digunakan untuk memecahkan masalah-masalah selain optimasi. Selama suatu masalah berbentuk adaptasi (alami maupun buatan), maka dapat diformulasikan dalam terminologi genetika.

Algoritma Genetika merupakan teknik *search stochastic* yang berdasarkan mekanisme seleksi alam dan genetika natural. Pada algoritma genetika, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin dikenal dengan istilah populasi. Setiap individu di dalam populasi disebut kromosom, yang merepresentasikan suatu penyelesaian terhadap masalah yang ditangani. Sebuah kromosom terdiri dari sebuah string yang berisi berbagai simbol, dan biasanya, tetapi tidak mutlak, string tersebut berupa sederetan bit-bit biner “0” dan “1”. Sebuah kromosom tumbuh atau berkembang biak melalui berbagai iterasi yang berulang-ulang, dan disebut sebagai generasi. Pada setiap generasi, berbagai kromosom yang dihasilkan akan dievaluasi menggunakan suatu pengukuran fitness. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas dari kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah

anak (offspring) terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (parent) dengan menggunakan operator penyilangan (crossover). Selain operator penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai fitness dari kromosom induk (parent) dan nilai fitness dari kromosom anak (offspring), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik [15].

Secara skematis, siklus algoritma genetika dapat diilustrasikan seperti pada Gambar 2.22.



Gambar 2.22 Stuktur Umum *Algoritma Genetika* [14].

2.5.2 Sistem Operasi *Algoritma Genetika*

Terdapat sejumlah komponen utama yang terdapat dalam sistem operasi algoritma genetika, yaitu:

1. Teknik Penyandian (Pengkodean)
2. Prosedur Inisialisasi (Membangkitkan Populasi Awal)
3. Evaluasi Nilai Fitness
4. Seleksi Orang Tua
5. Rekombinasi
6. Mutasi
7. Elitism
8. Evaluasi Tingkat Keseragaman Unsur Kromosom

1. Teknik Penyandian (Pengkodean)

Teknik penyandian disini meliputi penyandian gen dari kromosom. Gen merupakan bagian dari kromosom, dimana satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk string bit, pohon, array bilangan real, daftar aturan, elemen permutasi, elemen program dan lain-lain. Contoh dari representasi kromosom antara lain sebagai berikut :

1. String bit : 10011, 11101, dst
2. Bilangan Real : 65.65, 562.88, dst
3. Elemen Permutasi : E2, E10, dst
4. Daftar Aturan : R1, R2, R3, dst
5. Elemen Program : pemrograman genetika, dst
6. Struktur lainnya

Misalkan ingin dipecahkan masalah estimasi fungsi produksi *Cobb-Dauglas* yaitu $y = \beta_1 L^{\beta_2} K^{\beta_3}$ dengan sampel yang ada untuk L dan K berapa nilai $\beta_1, \beta_2, \beta_3$ dengan fungsi tujuan meminimumkan *least square* atau memaksimumkan fungsi likelihood. Persoalan tersebut dapat diselesaikan dengan *Algoritma Genetika*, yaitu ketiga parameter $\beta_1, \beta_2, \beta_3$ dikodekan dalam kromosom yang masing-masing berisi sejumlah gen yang mengkodekan informasi yang disimpan di dalam kromosom. Misalkan untuk memudahkan digunakan binary encoding dengan

panjang kromosom 12 gen (12 bits), masing-masing parameter $\beta_1, \beta_2, \beta_3$ dikodekan dengan 4 gen, sehingga diperoleh pengkodean seperti pada tabel 2.4.

Tabel 2.4 Skema *Binary Encoding*

| Parameter | β_1 | | | | β_2 | | | | β_3 | | | |
|----------------|-----------|----|----|----|-----------|----|----|----|-----------|-----|-----|-----|
| Binary Number | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| | g1 | g2 | g3 | g4 | g5 | g6 | g7 | g8 | g9 | g10 | g11 | g12 |
| Decimal Number | 11 | | | | 14 | | | | 3 | | | |

Jika nilai parameter yang akan dicari mempunyai constraint, $\alpha < \beta < b$, maka berdasarkan *binary encoding* nilai parameter dapat diperoleh dengan menggunakan formula berikut:

$$\beta = \alpha + \beta_{dec} * \frac{b-a}{2^n-1} \quad (2.48)$$

di mana n menyatakan banyaknya bit atau gen (dalam tabel 2.4 setiap parameter memiliki empat 4 bit dan constraint $0 < \beta < 1$), sehingga diperoleh:

$$\begin{aligned} \beta_2 &= 0 + 11 * \frac{1-0}{2^4-1} = 0.7333 \\ \beta_2 &= 0 + 14 * \frac{1-0}{2^4-1} = 0.9333 \\ \beta_3 &= 0 + 3 * \frac{1-0}{2^4-1} = 0.2 \end{aligned} \quad (2.49)$$

Setelah skema pengkodean ditentukan, *Algoritma Genetika* diinisialisasi untuk sebuah populasi dengan N kromosom. Gen-gen yang mengisi masing-masing kromosom dibangkitkan secara random. Masing-masing kromosom akan dikodekan menjadi individu dengan nilai *fitness* tertentu, dan kemudian sebuah populasi baru akan dibentuk dengan menggunakan mekanisme seleksi alamiah, yaitu memilih individu-individu secara proporsional terhadap nilai *fitness*-nya, dan genetika alamiah, yakni pindah silang (*crossover*) serta mutasi. Pada algoritma genetika metode yang akan digunakan adalah dengan skema pergantian populasi yang disebut *generational replacement*, artinya, N kromosom dari suatu generasi digantikan sekaligus oleh N kromosom baru hasil pindah silang dan mutasi.

2. Prosedur Inisialisasi (Membangkitkan Populasi Awal)

Membangkitkan populasi awal adalah membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

Teknik dalam membangkitkan populasi awal ini ada beberapa macam, di antaranya adalah sebagai berikut:

- 1) Random Generator
- 2) Pendekatan tertentu (memasukan nilai tertentu kedalam gen)
- 3) Permutasi gen

3. Evaluasi Nilai *Fitness*

Ada tiga langkah dalam proses mengevaluasi nilai *fitness* kromosom, yaitu:

1. Mengganti genotip kromosom menjadi fenotip kromosom, ini berarti mengganti binary strings menjadi *real value*
2. Mengevaluasi fungsi objektif
3. Mengganti nilai dari fungsi objektif menjadi nilai *fitness*. Agar nilai *fitness* selalu bernilai positif, maka nilai *fitness* dari setiap kromosom sama dengan memaksimumkan objektif dikurangi objektif yang telah dievaluasi untuk setiap kromosom dalam populasi.

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Di dalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup, sedangkan individu yang bernilai *fitness* rendah akan mati. Pada masalah optimasi dalam tugas ini, solusi yang akan dicari adalah memaksimumkan sebuah fungsi *likelihood* dan meminimumkan *least square* fungsi produksi *Cobb-Dauglas* dan fungsi produksi CES.

4. Seleksi Orang Tua

Pemilihan dua buah kromosom yang dijadikan induk atau sebagai orang tua dilakukan secara proporsional sesuai dengan dengan nilai *fitness*-nya. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung dari nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap seleksi berikutnya.

Pada proses seleksi orang tua, metode yang paling umum digunakan adalah *roulette wheel selection*. Metode seleksi roda *roulette* ini merupakan metode yang paling sederhana serta paling banyak digunakan, dan sering juga dikenal dengan nama *stochastic sampling with replacement*. Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara beraturan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan dengan ukuran *fitness*-nya. Sebuah bilangan random akan dibangkitkan dan individu yang memiliki segmen dalam kawasan bilangan random tersebut akan diseleksi. Proses ini diulang hingga diperoleh sejumlah individu yang diharapkan. Skema dengan seleksi roda *roulette* ini adalah berdasarkan *fitness scale*. Terpilihnya suatu kromosom dalam populasi untuk dapat berkembang biak sebanding dengan *fitness*-nya. *Tradeoff* antara eksplorasi dan eksploitasi terjadi jika terdapat satu atau sekelompok kecil kromosom yang mempunyai *fitness* yang baik, yaitu mengeksplorasi bagian-bagian baru dalam ruang pencarian, atau terus mengeksplorasi informasi yang telah diperoleh. Kecenderungan kromosom yang baik untuk terpelihara terus dapat membawa ke hasil optimum lokal atau konvergensi dini (*premature convergence*) ke suatu hasil yang bukan optimum global. Namun demikian, jika semua kromosom dalam populasi mempunyai *fitness* yang hampir sama, maka seleksi ini akan menjadi seleksi yang bersifat acak.

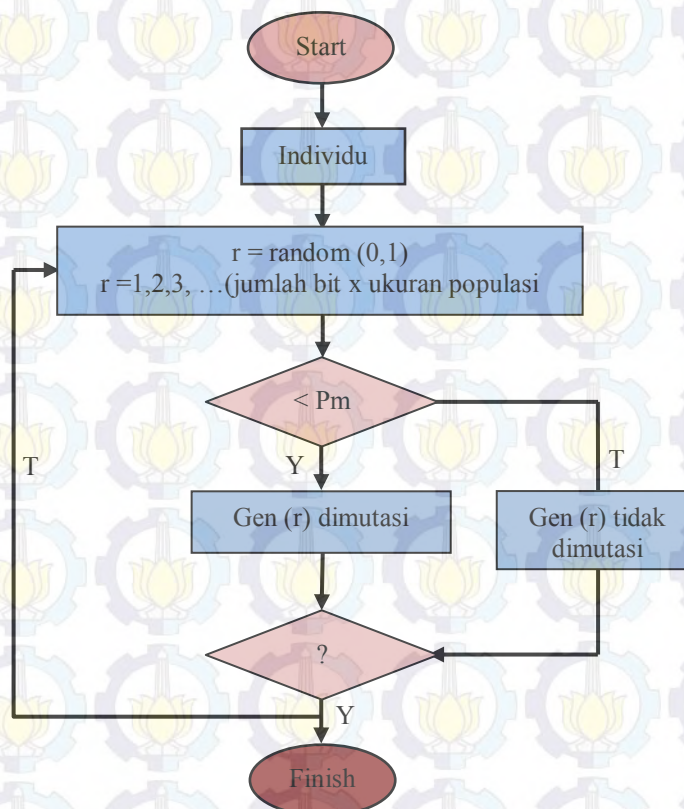
5. Rekombinasi

Algoritma genetika merupakan proses pencarian yang *heuristic* dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan algoritma genetika dalam menemukan solusi optimum suatu masalah yang diberikan. Hal yang harus diperhatikan adalah menghindari terjadinya

konvergensi prematur, di mana dicapai solusi optimum yang belum waktunya, dalam arti bahwa solusi yang diperoleh adalah hasil optimum lokal.

6. Mutasi

Mutasi merupakan proses untuk mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operasi *crossover* yang dilakukan pada kromosom dengan tujuan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih baik, dan lama-kelamaan menuju solusi optimum yang diinginkan. Akan tetapi, untuk mencapai hal ini, penekanan selektif juga memegang peranan yang penting. Jika dalam proses pemilihan kromosom-kromosom cenderung terus pada kromosom yang memiliki *fitness* yang tinggi saja, konvergensi prematur akan sangat mudah terjadi. Secara skematis proses mutasi dapat ditunjukkan pada Gambar 2.23.



Gambar 2.23 Proses Mutasi Pada *Algoritma Genetika* [14].

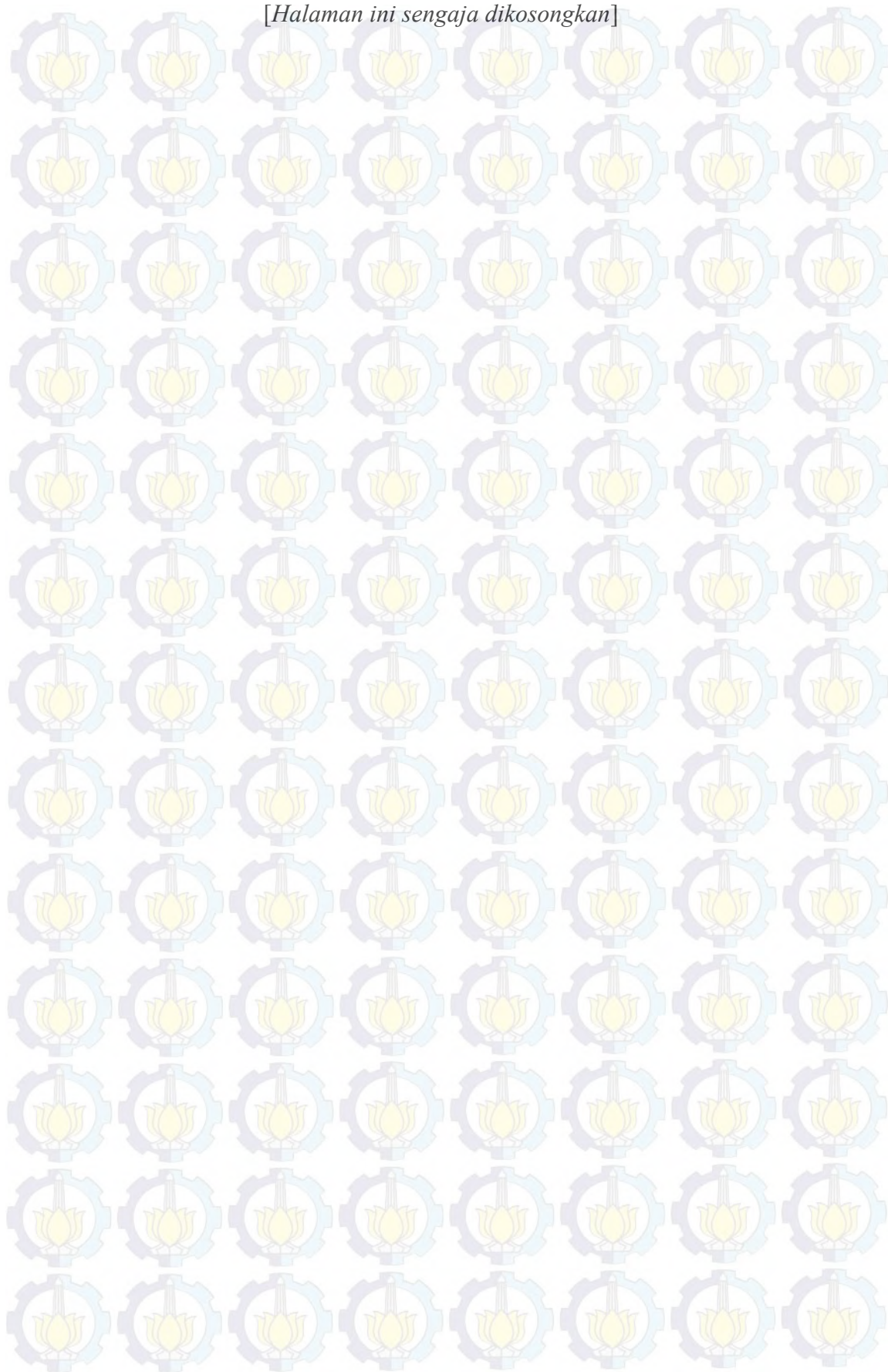
7. Elitism

Proses seleksi dilakukan secara random sehingga tidak ada jaminan bahwa suatu individu yang bernilai *fitness* tertinggi akan selalu terpilih. Walaupun individu bernilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai *fitness*-nya menurun) karena proses pindah silang. Oleh karena itu, untuk menjaga agar individu bernilai *fitness* tertinggi tersebut tidak hilang selama evolusi, maka perlu dibuat satu atau beberapa kopinya. Prosedur ini dikenal sebagai elitisme.

8. Evaluasi Tingkat Keseragaman Unsur Kromosom

Generasi terbaik pada dasarnya adalah representasi hasil nilai optimasi fungsi objektif. Generasi ini akan ditunjukkan dengan memiliki tingkat keseragaman kromosom yang tinggi untuk semua populasi yang ada. Jika proses evolusi terus berlangsung dan telah dibuktikan bahwa secara matematis proses dalam algoritma genetika akan menghasilkan generasi terbaik yang memiliki *fitness* yang tinggi, maka bisa diduga bahwa generasi tersebut akan memiliki tingkat keseragaman unsur kromosom yang tinggi. Karena hanya populasi yang memiliki sifat-sifat yang *fit* dengan *objective function* saja yang dapat *survive* dan berkembang biak. Semakin tinggi tingkat keseragaman menunjukkan bahwa populasi dalam suatu generasi memiliki sifat serupa, yang ditunjukkan dengan susunan gen dalam kromosomnya mirip pada seluruh populasi yang ada.

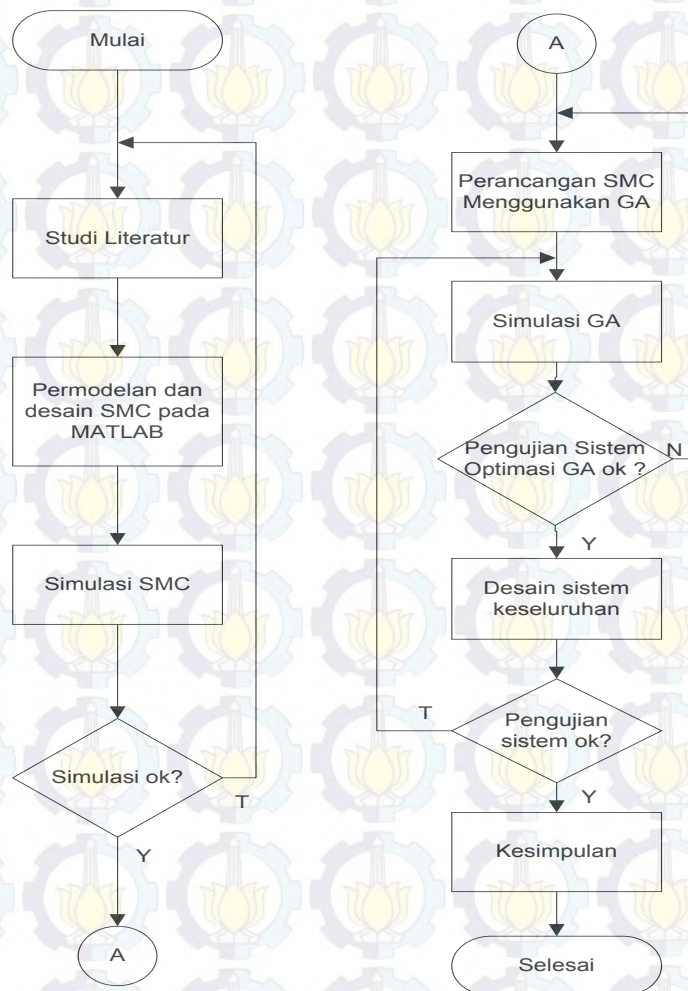
[Halaman ini sengaja dikosongkan]



BAB III

PERANCANG SISTEM

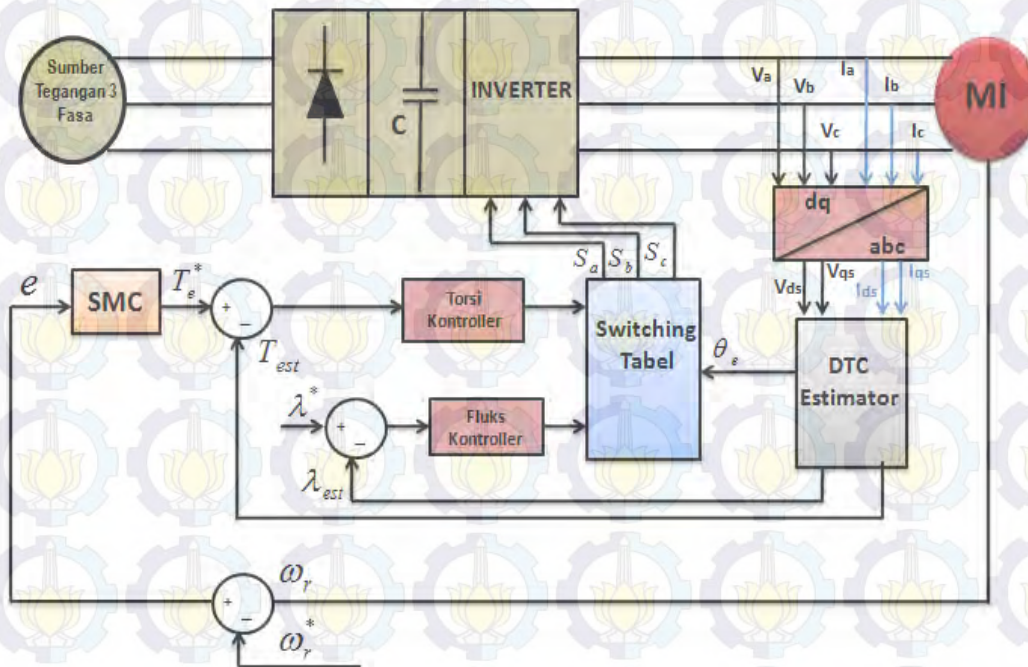
Pada bab ini akan dibahas mengenai tahapan-tahapan yang dilakukan dalam proses perancangan sistem secara detail. Proses perancangan yang dilakukan meliputi proses perancangan simulasi dari model matematika motor induksi, perancangan simulasi *Direct Torque Control* (DTC), perancangan simulasi pemodelan identifikasi motor induksi dengan DTC, dan yang terakhir yaitu perancangan simulasi kontroler *Sliding Mode Control* (SMC) Berbasis *Algoritma Genetika*. Seluruh perancangan yang dilakukan menggunakan program *Simulink* MATLAB. Alur pengerjaan perancangan sistem dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Perancangan Sistem Keseluruhan

3.1 Gambaran Umum Sistem

Secara keseluruhan sistem dapat digambarkan dalam diagram blok agar mudah untuk memahami cara kerja dan bagian-bagian sistem yang ditunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram Blok Sistem Keseluruhan

Gambar diagram blok di atas menunjukkan bahwa, keluaran inverter berupa tegangan tiga fasa sebagai masukan untuk motor induksi kemudian tegangan dan arus stator pada motor induksi yaitu V_a , V_b , V_c dan I_a , I_b , I_c akan ditransformasi oleh blok abc ke dq menggunakan persamaan (3.1) dengan maksud untuk mengubah koordinat motor induksi dari 3 fasa ke dalam koordinat 2 fasa. Hasil dari transformasi berupa tegangan d_s , q_s dan arus d_s , q_s inilah sebagai masukan untuk blok DTC estimator, sehingga keluaran dari DTC estimator yaitu torsi estimator akan dibandingkan dengan torsi error referensi (torsi hasil kontroler SMC-GA) dan fluks estimator akan dibandingkan dengan fluks referensi sedangkan sudut fluks stator sebagai masukan untuk blok *switching table* gunanya untuk mengontrol fluks stator dan torsi untuk memenuhi nilai referensi yang telah ditentukan. Selanjutnya putaran motor akan dibandingkan dengan putaran

referensi dan hasil perbandingan inilah nantinya akan dikontrol menggunakan metode SMC-GA.

3.2 Perancangan Simulasi Model Motor Induksi

Tahapan awal dalam melakukan perancangan sistem setelah melakukan studi literatur adalah mendapatkan model matematis dari *plant* yaitu motor induksi, agar *plant* tersebut dapat disimulasikan pada *software* yang digunakan dan untuk mengetahui karakteristik dari *plant*. Model matematis dari motor induksi yang digunakan dalam perancangan ini didapatkan dari persamaan (2.29) yaitu:

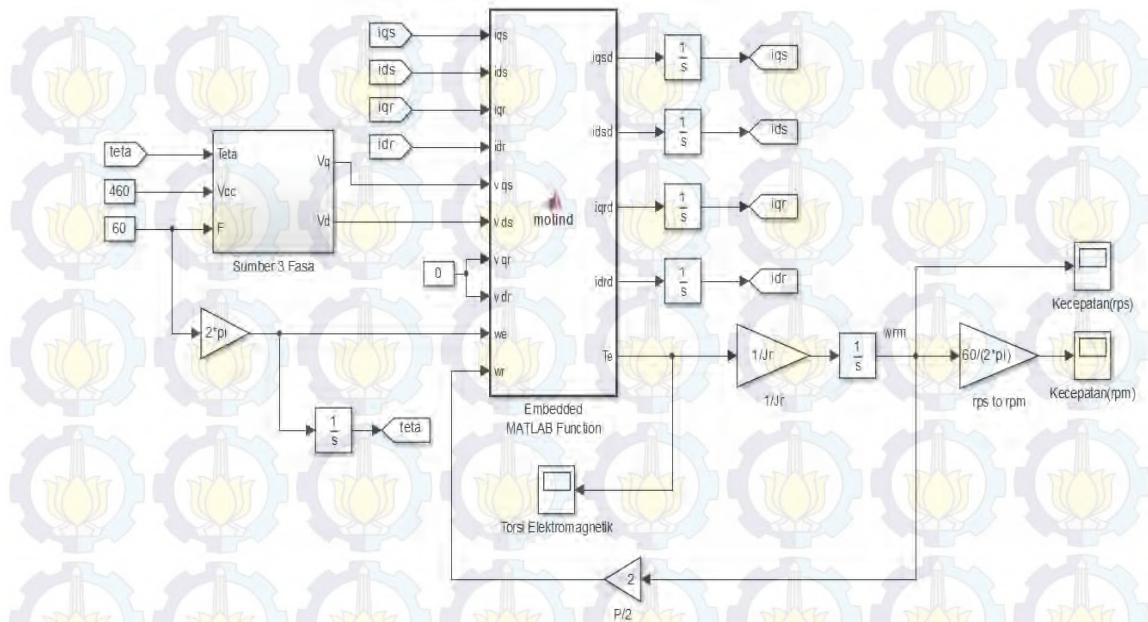
$$\frac{d}{dt} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} = \mathbf{Q}^{-1} \begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} - \mathbf{Q}^{-1} \mathbf{P} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix}$$

Kemudian persamaan (2.29) akan digunakan untuk mencari nilai arus dari motor induksi. Dalam model matematis motor induksi terdapat nilai tahanan dan induktansi yang harus diketahui sebagai parameter. Selain nilai tahanan dan induktansi, pada perancangan simulasi motor induksi diperlukan parameter-parameter motor induksi lainnya seperti nilai inertia, jumlah kutub, dan lain-lain. Parameter motor induksi yang digunakan pada perancangan dan simulasi dapat dilihat pada Tabel 3.1.

Tabel 3.1 Parameter Motor Induksi 3 HP [11].

| No. | Nama (Satuan) | Nilai |
|-----|--------------------------------------|----------|
| 1. | Tenaga motor (HP) | 3/2,4 kW |
| 2. | Tegangan motor (line to line) (Volt) | 460 |
| 3. | Frekuensi (Hz) | 60 |
| 4. | Jumlah kutub | 4 |
| 5. | Tahanan stator (Ohm) | 1,77 |
| 6. | Tahanan rotor (Ohm) | 1,34 |
| 7. | Induktansi stator (mH) | 0,3829 |
| 8. | Induktansi rotor (mH) | 0,3811 |
| 9. | Induktansi magnetic (mH) | 0,369 |
| 10. | Momen inersia (kg.m^2) | 0,025 |
| 11. | Arus beban penuh (A) | 4 |
| 12. | Kecepatan beban penuh (RPM) | 1750 |
| 13. | Efisiensi beban penuh (%) | 88,5 |
| 14. | Faktor daya (%) | 80 |
| 15. | Slip beban penuh (%) | 1,72 |

Setelah mengetahui persamaan model matematis dari *plant* (motor induksi) dan parameter dari motor induksi, tahap selanjutnya yaitu menyajikan persamaan model matematis tersebut dalam bentuk fungsi dan *simulink* MATLAB. Perancangan simulasi motor induksi yang dibuat dapat dilihat pada Gambar 3.3. berikut ini:



Gambar 3.3 Perancangan Simulasi Model Motor Induksi

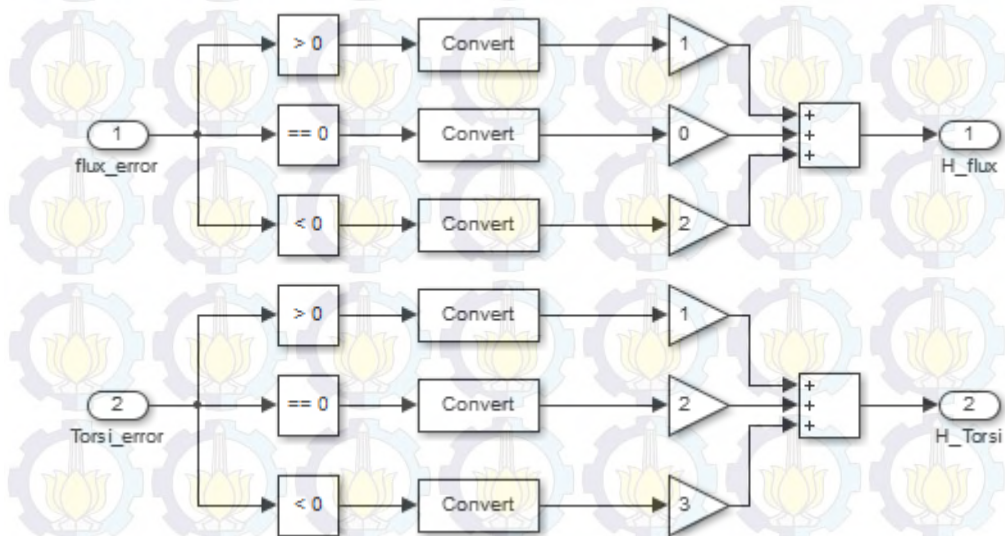
Pada Gambar 3.3. dapat dilihat bahwa simulasi yang dirancang secara garis besar terdiri dari dua blok salah satunya blok *Embedded MATLAB Function* yang berfungsi untuk memasukkan fungsi dari persamaan model matematis dan parameter dari motor induksi .

3.3 Perancangan *Direct Torque Control (DTC)* pada Simulasi Model Motor Induksi

Tahapan selanjutnya setelah membuat simulasi model motor induksi adalah membuat simulasi DTC pada model motor induksi yang sudah dibuat. Pada pembuatan simulasi DTC terdapat dua masukan yang digunakan yaitu nilai fluks acuan dan torsi acuan. Cara kerja DTC ini yaitu mempertahankan nilai keluaran fluks dan torsi agar sesuai dengan nilai acuan yang diberikan. Sesuai dengan Gambar 2.11, pada metode DTC terdapat bagian-bagaian yang menyusun sistem DTC tersebut yaitu komparator, *tabel switching*, estimator, dan sektor sudut fluks stator. Bagian-bagian tersebut akan dijelaskan bagaimana proses merancangnya menjadi bentuk simulasi dalam *software*.

3.3.1 Perancangan Kontroler Histerisis

Pada pembahasan sebelumnya kontroler histerisis digunakan untuk mendapatkan nilai kesalahan fluks dan torsi pada daerah histerisisnya. Nilai kesalahan (*error*) ini digunakan sebagai masukan komparator yang nantinya diolah menjadi nilai level histerisis, yaitu dua level histerisis untuk fluks dan tiga level histerisis untuk torsi. Hasil perancangan kontroler histerisis baik untuk fluks maupun torsi dapat dilihat pada Gambar 3.4.



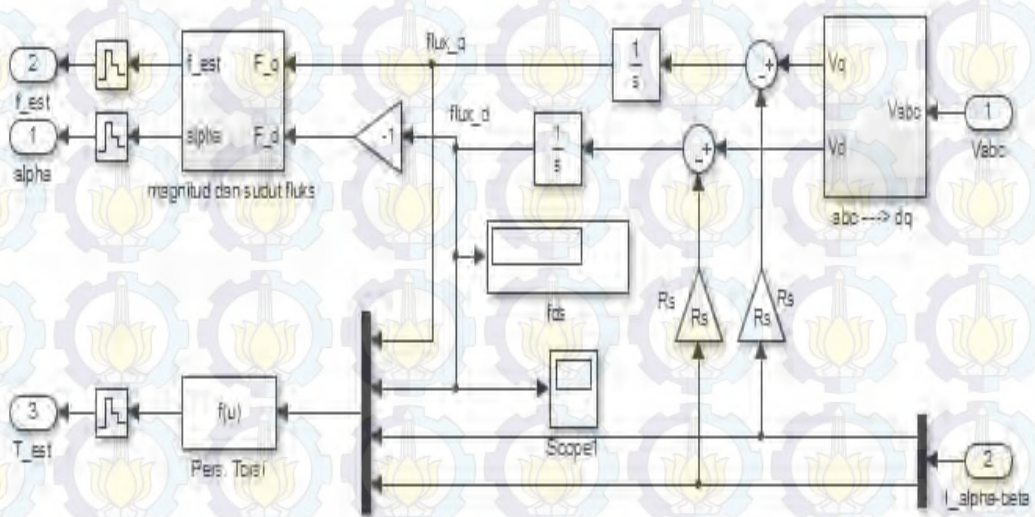
Gambar 3.4 Perancangan Kontroler Histerisis Fluks dan Torsi DTC

Dari Gambar 3.4. dapat dilihat bahwa setiap nilai kesalahan fluks ($\Delta\psi_s$) dan kesalahan torsi (ΔT_e) akan dibandingkan dengan nilai nol (0). Jika nilai kesalahan lebih besar daripada 0 maka nilai fluks dan torsi akan menghasilkan sinyal keluaran sebesar 1. Jika nilai kesalahan kurang dari 0 maka fluks akan menghasilkan sinyal keluaran sebesar 2 dan torsi akan menghasilkan sinyal keluaran sebesar 3. Kondisi terakhir jika nilai kesalahan fluks dan torsi bernilai 0 maka fluks akan menghasilkan sinyal keluaran 0 yang artinya motor akan berhenti sehingga nilai fluks stator tidak boleh sama dengan 0, sedangkan torsi akan menghasilkan sinyal keluaran sebesar 2.

3.3.2 Perancangan Estimator

Salah satu kelebihan dari DTC adalah bisa dilakukan tanpa menggunakan sensor kecepatan, dan sebagai gantinya maka digunakan blok estimator. Pada estimator ini dihitung besarnya nilai fluks dan torsi sebagai umpan balik nilai fluks dan torsi referensi. Selain itu dihitung juga besarnya sudut dari fluks stator (sektor). Nilai torsi estimasi dihitung dengan menggunakan persamaan (2.8), sedangkan nilai magnitud dan sudut fluks stator estimasi dihitung dengan menggunakan persamaan (2.31), dan (2.32).

Hasil perancangan estimator DTC dapat dilihat pada Gambar 3.5.

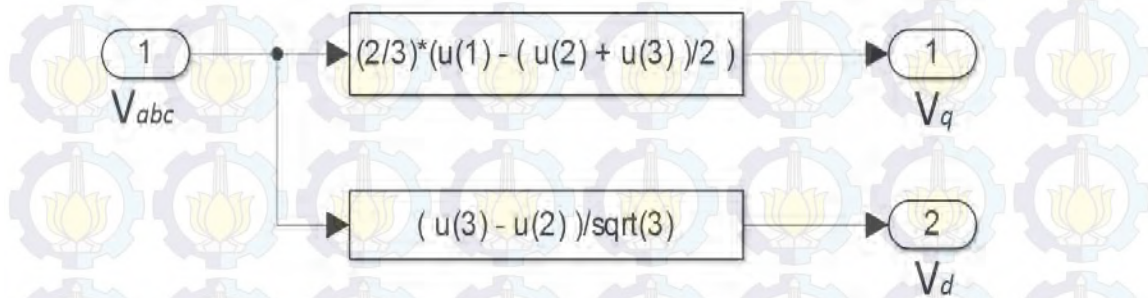


Gambar 3.5 Perancangan Estimator DTC

Pada Gambar 3.5. dapat dilihat bahwa estimator menerima sinyal masukan dari tegangan yang dihasilkan oleh inverter berupa tegangan tiga fasa dan arus 2 fasa dari motor induksi. Sinyal masukan tegangan tiga fasa harus diubah ke dalam tegangan 2 fasa karena dalam menentukan magnitud dan sudut fluks stator yang dibutuhkan adalah tegangan dalam frame dua fasa (d-q). Persamaan yang digunakan untuk mengubah tegangan tiga fasa menjadi tegangan dua fasa dapat dilihat pada persamaan (3.1).

$$\begin{bmatrix} V_q \\ V_d \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \quad (3.1)$$

Persamaan (3.1) digunakan untuk membuat blok ‘abc-->dq’ pada blok estimator dan dibuat dalam fungsi MATLAB untuk perancangan simulasinya seperti pada Gambar 3.6.



Gambar 3.6 Transformasi Tegangan Dari abc ke dq

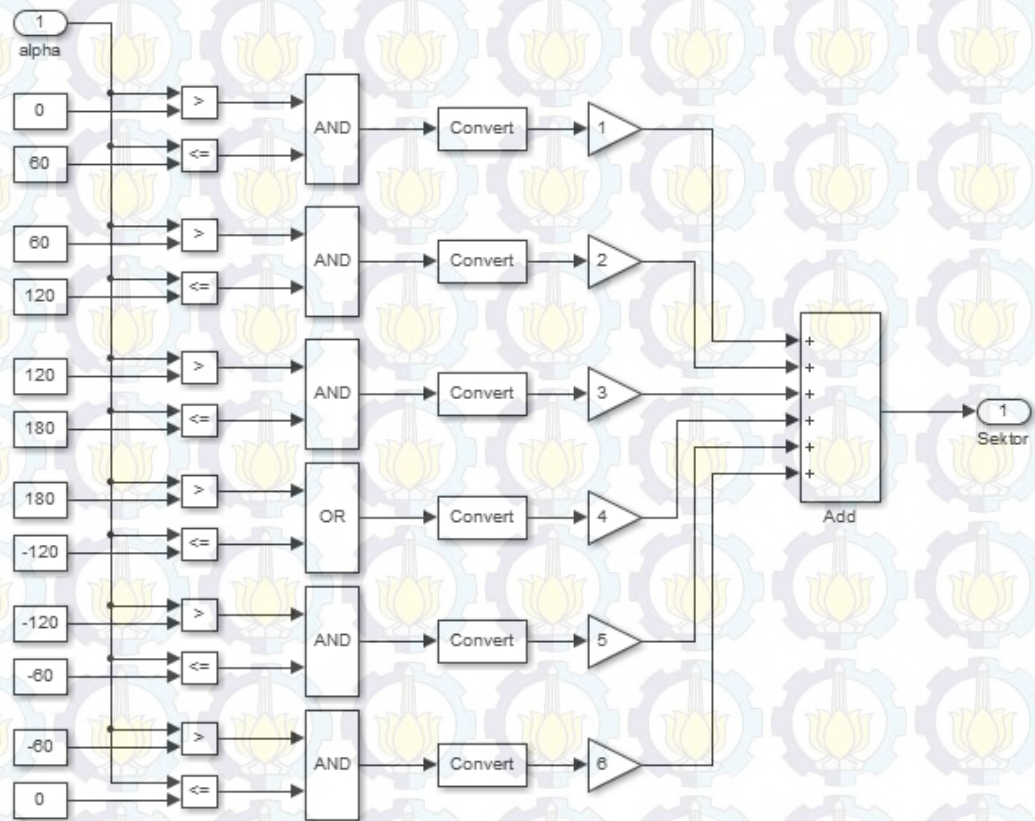
3.3.3 Perancangan Sektor Sudut Fluks Stator

Perancangan sektor sudut fluks stator bertujuan untuk mengubah besarnya sudut fluks stator (α) menjadi dalam bentuk sektor. Sektor sudut fluks stator dibagi menjadi enam dan masing-masing sektor besarnya 60° . Besarnya masing-masing sektor dapat dilihat pada Tabel 3.2.

Tabel 3.2 Sektor Sudut Fluks Stator

| No. | Sektor (Simbol) | Besar |
|-----|-------------------|----------------------------------|
| 1. | 1 ($\alpha(1)$) | $> 0^\circ - < 60^\circ$ |
| 2. | 2 ($\alpha(2)$) | $> 60^\circ - < 120^\circ$ |
| 3. | 3 ($\alpha(3)$) | $> 120^\circ - < 180^\circ$ |
| 4. | 4 ($\alpha(4)$) | $> 180^\circ - < (-120^\circ)$ |
| 5. | 5 ($\alpha(5)$) | $> (-120^\circ) - < (-60^\circ)$ |
| 6. | 6 ($\alpha(6)$) | $> (-60^\circ) - < 0^\circ$ |

Dari data Tabel 3.2. dapat dibuat rancangan simulasi sektor sudut fluks stator yang dapat dilihat pada Gambar 3.7.



Gambar 3.7 Perancangan Sektor Sudut Fluks

Dari Gambar 3.7 dapat dilihat bahwa sinyal masukan berupa besarnya sudut fluks stator yang dihasilkan estimator akan ditentukan letaknya sesuai sektornya. Nilai sudut fluks stator akan dibandingkan nilainya dengan besarnya sudut untuk masing-masing sektor. Sektor sudut fluks stator ditentukan dari posisi nilai sudut fluks stator itu berada sesuai dengan batas untuk masing-masing sektor pada Tabel 3.2. Sinyal keluaran berupa nilai satu sampai enam yang masing-masing mewakili nilai keenam sektor.

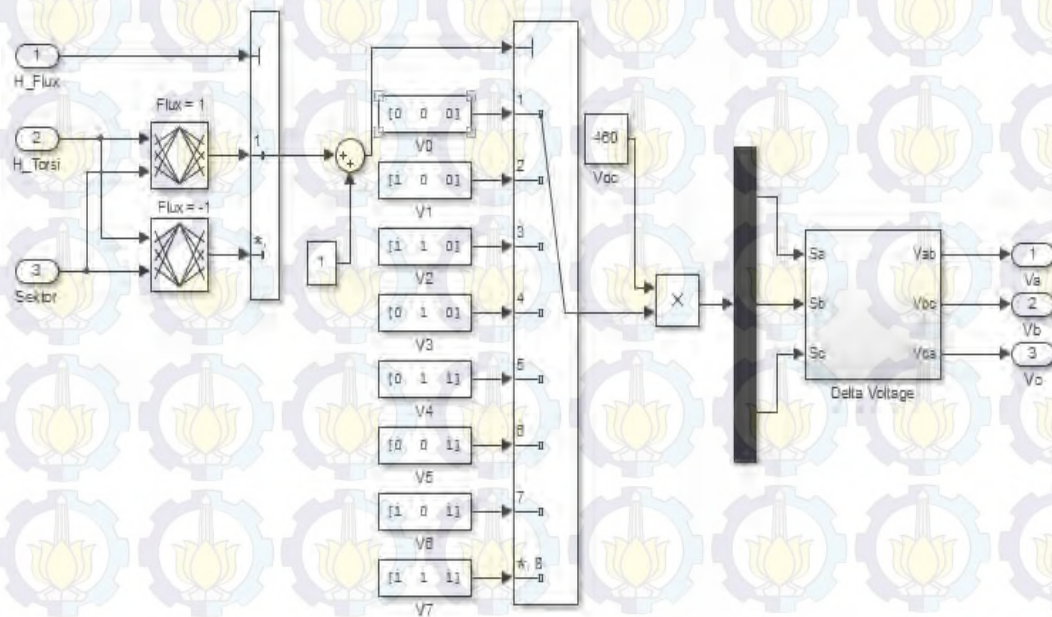
3.3.4 Perancangan Tabel *Switching*

Tiga faktor yang mempengaruhi tabel *switching* yaitu besarnya histerisis fluks stator, besarnya histerisis torsi dan sektor sudut fluks stator yang aktif. Nilai histerisis fluks stator dan torsi didapatkan dari sinyal keluaran komparator

sedangkan sektor sudut fluks didapatkan dari sinyal keluaran blok sektor sudut fluks stator.

Konsep utama dari DTC adalah untuk menghasilkan vektor tegangan yang aktif secara bergantian untuk masukan inverter sehingga tegangan masukan pada motor induksi merupakan sinyal sinusoidal. Dapat dikatakan bahwa tabel *switching* yang sangat menentukan vektor tegangan mana yang akan aktif secara bergantian. Tabel *switching* dapat dilihat pada Tabel 2.2, di mana pada tabel tersebut disajikan aturan vektor tegangan mana saja yang akan aktif sesuai dengan kondisi sinyal masukan.

Hasil perancangan simulasi dari data tabel *switching* dapat dilihat pada Gambar 3.8.

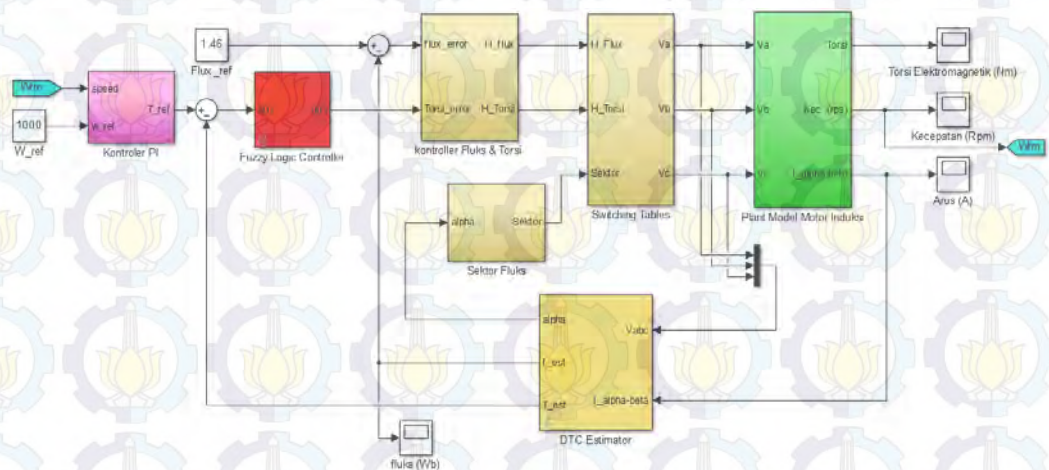


Gambar 3.8 Perancangan Tabel *Switching*

Berdasarkan Gambar 3.8. aturan pada Tabel 2.2. untuk menghasilkan vektor tegangan mana yang akan aktif dimasukkan pada blok '*look-up table*' yang terbagi menjadi dua, yaitu untuk kondisi ketika nilai fluks = 1 dan kondisi nilai fluks = (-1). Pada simulasi ini, sinyal keluaran yang dihasilkan berupa tegangan tiga fasa yang digunakan untuk menyuplai tegangan pada motor induksi dan sebagai masukan pula untuk blok estimator.

3.4 Perancangan Model Motor Induksi Dengan *Direct Torque Control* dan Kontroler Proporsional Integral (DTC-PI) Menggunakan *Fuzzy Logic Controller* (FLC)

Simulasi yang dilakukan merupakan gabungan dari perancangan model motor induksi ditambah dengan *Direct Torque Control* (DTC) pada model motor induksi tersebut. Selain itu ditambahkan juga kontroler *Proporsional Integral* (PI) dan *Fuzzy Logic Controller* (FLC) untuk mengatur masukan yang berupa nilai referensi kecepatan yang akan diubah menjadi torsi dan menjadi masukan untuk blok kontroler torsi histerisis. Proses simulasi yang dilakukan merupakan tahapan validasi dari model motor induksi yang dihubungkan dengan *Direct Torque Control* (DTC), apakah model motor induksi dapat terintegrasi dengan baik dengan *Direct Torque Control* (DTC) dan dapat memberikan respon sesuai dengan nilai acuan/referensi yang diberikan pada kontroler PI dan FLC. Dan proses simulasi ini juga dilakukan hanya sebatas perbandingan dengan apa yang ingin diteliti, apakah ada perbedaan dari setiap respon sistem. Gambar 3.9. merupakan simulasi model motor induksi dengan DTC-PI menggunakan *Fuzzy Logic Controller* (FLC).



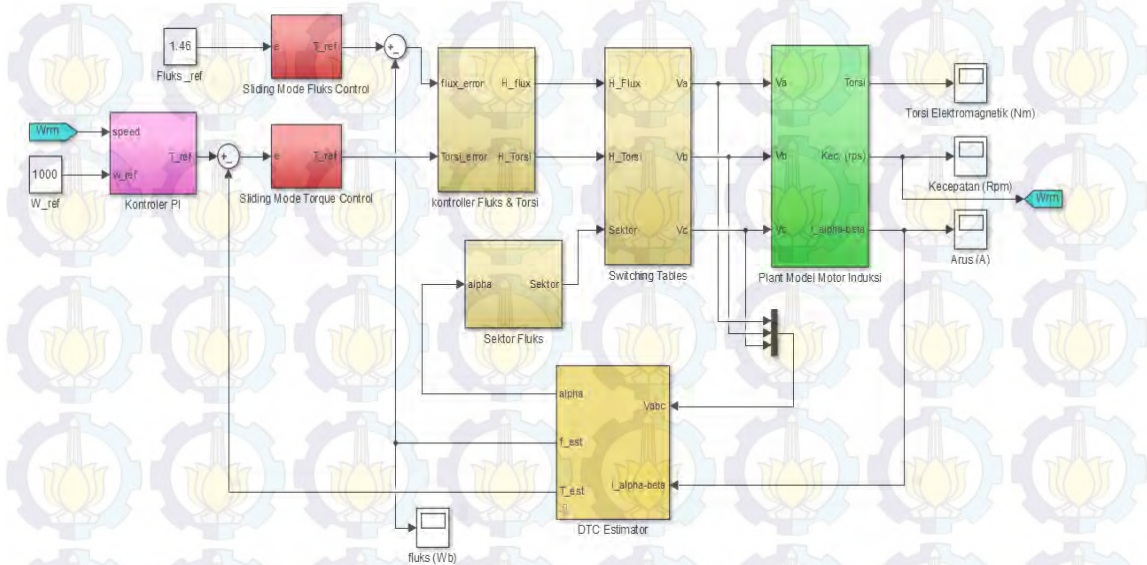
Gambar 3.9 Simulasi Model Motor Induksi dengan DTC-PI Menggunakan *Fuzzy Logic Controller* (FLC).

Dari Gambar 3.9. dapat dilihat bahwa simulasi terdiri dari beberapa sub-sistem, yaitu sub-sistem yang berwarna ungu menunjukkan kontroler PI, sub-sistem yang berwarna merah adalah kontroler *Fuzzy Logic Controller* (FLC), sub-

sistem yang berwarna kuning kecoklatan menunjukkan DTC, dan sub-sistem yang berwarna hijau menunjukkan model *plant* motor induksi. Nilai acuan/referensi untuk fluks stator ditentukan sebesar 1.46 wb, sedangkan nilai acuan/referensi untuk kecepatan sebesar 1000 rpm. Nilai referensi kecepatan inilah yang nantinya dijadikan acuan untuk nilai respon keluaran sistem yang diinginkan.

3.5 Perancangan Model Motor Induksi Dengan *Direct Torque Control* dan Kontroler Proporsional Integral (DTC-PI) Menggunakan *Sliding Mode Control* (SMC)

Jika pada simulasi sebelumnya menggunakan metode *Fuzzy Logic Controller*, maka pada simulasi ini digunakan metode *Sliding Mode Control* (SMC) sebagai pengaturan kecepatan motor induksi. Simulasi ini merupakan gabungan dari *Direct Torque Control* dan Proporsional Integral (DTC-PI) menggunakan *Sliding Mode Control* (SMC). Proses simulasi yang dilakukan ini juga sebagai bentuk perbandingan dengan apa yang ingin diteliti. Pada simulasi ini digunakan kontroler terpisah antara fluks dan torsi. Gambar 3.10. simulasi model motor induksi dengan DTC-PI menggunakan *Sliding Mode Control* (SMC).



Gambar 3.10 Simulasi Model Motor Induksi dengan DTC-PI Menggunakan *Sliding Mode Control* (SMC).

Dari Gambar 3.10. dapat dilihat bahwa simulasi terdiri dari beberapa sub-sistem, yaitu sub-sistem yang berwarna ungu menunjukkan kontroler PI, sub-sistem yang berwarna merah adalah kontroler *Sliding Mode Controller* (SMC), sub-sistem yang berwarna kuning kecoklatan menunjukkan DTC, dan sub-sistem yang berwarna hijau menunjukkan model *plant* motor induksi. Nilai acuan/referensi untuk fluks stator ditentukan sebesar 1.46 wb, sedangkan nilai acuan/referensi untuk kecepatan sebesar 1000 rpm. Nilai referensi kecepatan inilah yang nantinya dijadikan acuan untuk nilai respon keluaran sistem yang diinginkan.

3.6 Perancangan Kontroler *Sliding Mode Control* (SMC)

Kontroler dirancang untuk mengatasi masalah pengaturan kecepatan motor induksi 3 fasa. Metode yang diajukan adalah metode kontrol *Sliding Mode Control* (SMC) berbasis *Algoritma Genetika*, dimana *Algoritma Genetika* dirancang untuk masalah optimalisasi parameter *Sliding Mode Control* (SMC).

Langkah awal untuk mendapatkan persamaan persamaan *Sliding Mode Control* (SMC) adalah dengan mengambil hukum kontrol Lyapunov yang didefinisikan sebagai berikut:

$$V_i = \frac{1}{2}s^2 \quad (3.2)$$

turunan pertama fungsi Lyapunov adalah

$$\dot{V}_i = s^T \dot{s} < 0 \quad (3.3)$$

di mana s adalah *sliding surface* atau permukaan luncur.

Sliding Mode Control terdiri dari penjumlahan dua sinyal kontrol, yang didefinisikan sebagai berikut:

$$u = u_n + u_{eq} \quad (3.4)$$

di mana u_{eq} adalah sinyal kontrol ekuivalen yang berperan untuk memindahkan *state* ke titik referensi, dan sinyal kontrol natural u_n berperan untuk menjaga sistem tetap berada pada referensi dengan permukaan luncur $s=0$.

Untuk mendapatkan sinyal kontrol ekuivalen, definisikan *error* putaran rotor yang merupakan beda antara posisi wahana dengan target dapat ditulis dengan persamaan sebagai berikut:

$$e = \omega - \omega_{ref} \quad (3.5)$$

dengan ω_{ref} adalah sinyal referensi. Langkah berikutnya adalah menentukan fungsi permukaan luncur (*Sliding Surface*), permukaan luncur didefinisikan oleh

$$s = \dot{e} + \lambda e \quad (3.6)$$

Untuk membuat permukaan luncur menuju 0 pada waktu tak hingga, maka diturunkan s terhadap waktu, didapatkan persamaan:

$$\dot{s} = \ddot{e} + \lambda \dot{e} = 0 \quad (3.7)$$

Substitusi e dengan Persamaan (3.5) maka akan didapatkan persamaan:

$$\dot{s} = \ddot{\omega} - \ddot{\omega}_{ref} + \lambda \dot{e} = 0 \quad (3.8)$$

Turunan pertama untuk mendapatkan $\dot{\omega}_r$:

$$\omega_r \frac{du}{dt} = \dot{\omega}_r \quad (3.9)$$

Kemudian turunan kedua untuk mendapatkan $\ddot{\omega}_r$:

$$\dot{\omega}_r \frac{du}{dt} = \ddot{\omega}_r \quad (3.10)$$

Turunan ketiga untuk mendapatkan \dot{e} yaitu dengan persamaan (3.11)

$$e \frac{du}{dt} = \dot{e} \quad (3.11)$$

Untuk menentukan sinyal kontrol ekuivalen u_{eq} didapatkan persamaan:

$$u_{eq} = \left(\ddot{\omega}_r - \dot{e} \lambda \left(\frac{1}{100s+1} \right) \right) J_{eq} \quad (3.12)$$

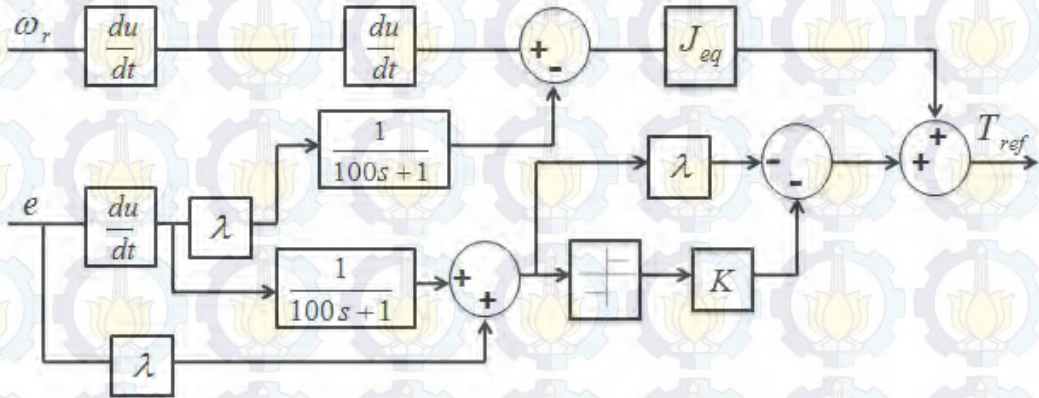
Selanjutnya menentukan sinyal kontrol natural u_n dapat ditulis dengan persamaan:

$$u_n = \left(K \cdot \text{sign} \left(\lambda e + \dot{e} \left(\frac{1}{100s+1} \right) \right) - \lambda \right) \quad (3.13)$$

sehingga jika dijumlahkan kedua sinyal kontrol tersebut untuk mendapatkan sinyal referensi atau T_{ref} sebagai masukan untuk sistem maka didapatkan persamaan (3.14) sebagai berikut:

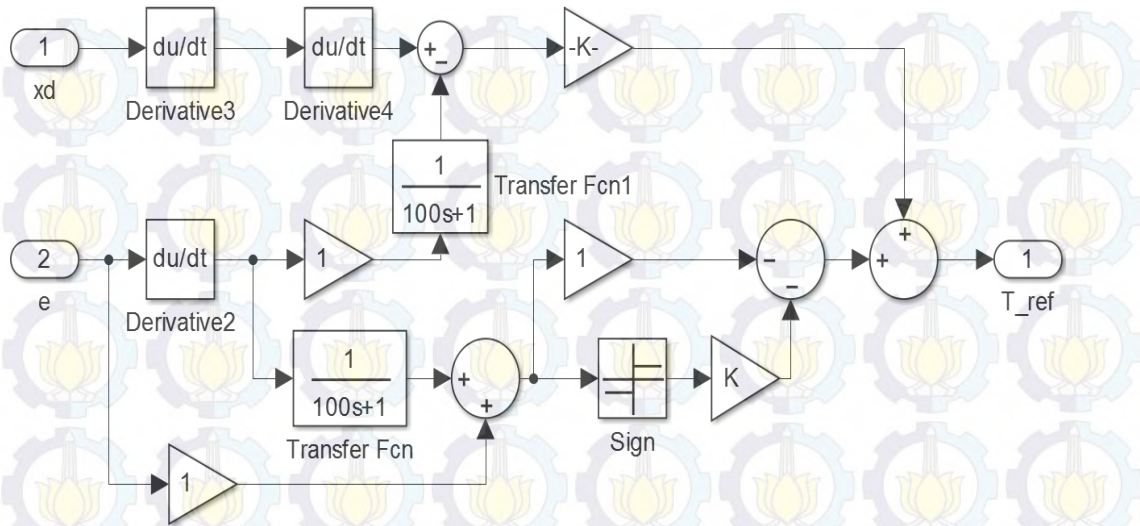
$$T_{ref} = \left(\ddot{\omega}_r - \dot{e} \lambda \left(\frac{1}{100s+1} \right) \right) J_{eq} + \left(K \cdot \text{sign} \left(\lambda e + \dot{e} \left(\frac{1}{100s+1} \right) \right) - \lambda \right) \quad (3.14)$$

Dengan persamaan (3.14) jika dibuat dalam bentuk blok diagram dapat dilihat seperti pada Gambar 3.11.



Gambar 3.11 Diagram Blok Kontroler *Sliding Mode Control* (SMC)

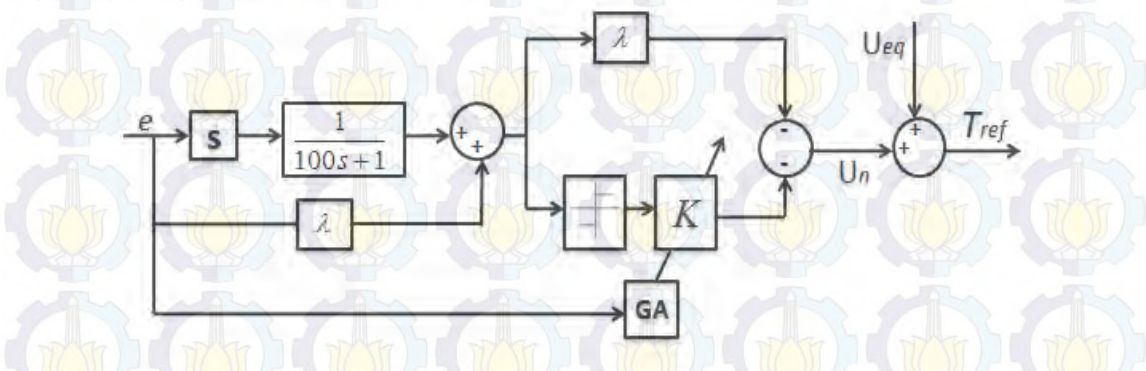
Kemudian persamaan (3.11) digunakan untuk membuat blok *Sliding Mode Control* (SMC) dibuat dalam fungsi MATLAB untuk perancangan simulasinya seperti pada Gambar 3.12.



Gambar 3.12 Perancangan Model Kontroler *Sliding Mode Control* (SMC)

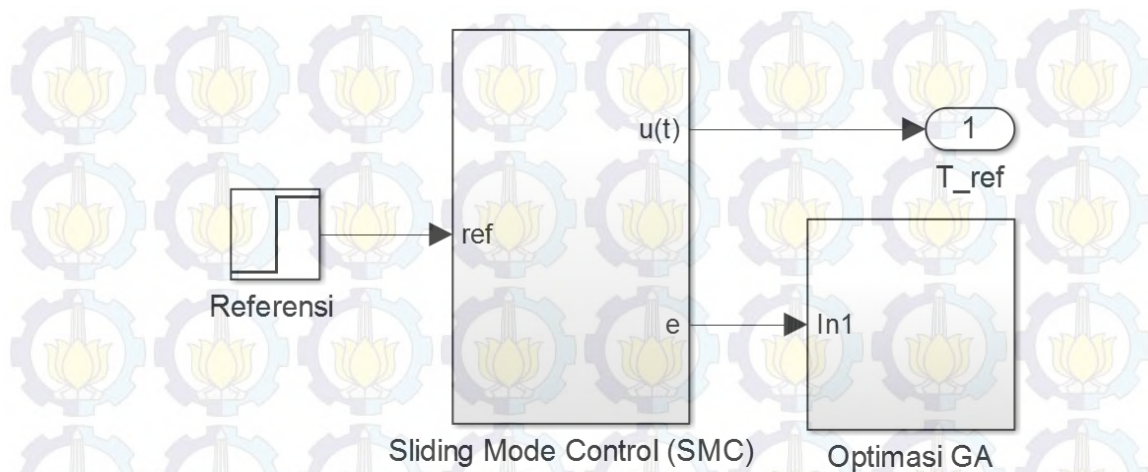
Berdasarkan Gambar 3.12. nilai error hasil dari perbandingan putaran rotor dan referensi sebagai masukan pada kontroler *Sliding Mode Control* (SMC) yang didefinisikan sebagai $e = \omega - \omega_{ref}$ (3.5).

Berdasarkan Gambar diagram blok di atas dapat disederhanakan seperti yang terlihat pada gambar 3.13.



Gambar 3.13 Diagram Blok Optimasi *Algoritma Genetika* Pada Parameter SMC.

Gambar 3.13. diagram blok di atas, dapat dijelaskan bahwa *error* hasil perbandingan putaran motor dengan referensi adalah sebagai input untuk *Algoritma Genetika* kemudian *error* tersebut dievaluasi untuk optimasi parameter *Sliding Mode Control* (SMC) yaitu *gain K*. Proses optimasi ini dilakukan secara *off-line*. Gambar 3.13. dibuat dalam fungsi MATLAB untuk perancangan simulasinya ditunjukkan seperti Gambar 3.14.

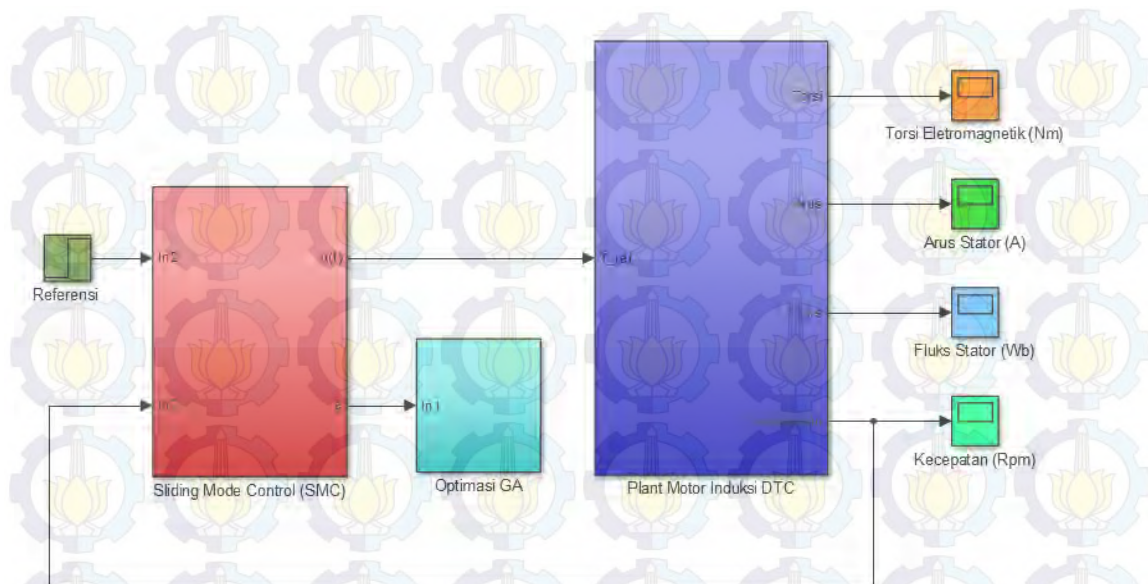


Gambar 3.14 Model Simulink Optimasi Menggunakan *Algoritma Genetika*.

Gambar 3.14 dapat dilihat bahwa, terdapat dua sub-sistem, sub-sistem yang pertama adalah rangkaian kontroler SMC dan sub-sistem yang kedua adalah optimasi *Algoritma Genetika* yang berfungsi untuk memasukkan sebuah fungsi solusi dari *Algoritma Genetika* mencari solusi suatu permasalahan dengan langkah awal membangkitkan sebuah populasi.

3.7 Perancangan Kontroler SMC *Algoritma Genetika* dengan *Plant Model Motor Induksi*

Blok kontroler perancangan evaluasi performansi error sudah didapatkan pada perancangan kontroler *Sliding Mode Control* (SMC) maka akan digunakan sebagai kontroler pada *plant* model motor induksi dengan *Direct Torque Control* (DTC). Hasil perancangan kontroler SMC dengan *plant* model motor induksi dengan *Direct Torque Control* (DTC) dapat dilihat pada Gambar 3.15.



Gambar 3.15 Perancangan Kontroler SMC *Algoritma Genetika* dengan *Plant* Model Motor Induksi

Dari Gambar 3.15. dapat dilihat bahwa, secara garis besar simulasi terdiri dari tiga sub-sistem, ketiga sub-sistem ini adalah hasil penyederhanaan dari beberapa sub-sistem. Sub-sistem yang berwarna merah adalah kontroler *Sliding Mode Control* (SMC), sedangkan sub-sistem yang berwarna biru muda adalah optimalisasi *Algoritma Genetika* dan yang berwarna biru tua adalah *Plant* Motor Induksi dengan *Direct Torque Control* (DTC). Pada simulasi ini, nilai acuan/referensi untuk fluks stator ditentukan sebesar 1.46 wb, sedangkan nilai acuan/referensi untuk kecepatan sebesar 1000 rpm. Nilai referensi kecepatan inilah yang nantinya dijadikan acuan untuk nilai respon keluaran sistem yang diinginkan. Adapun respon yang akan diamati adalah kecepatan, torsi elektromagnetik, arus stator dan fluks stator.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai simulasi dari hasil perancangan sistem pada BAB III. Setelah dilakukan simulasi kemudian dilakukan analisis terhadap data hasil simulasi apakah sesuai dengan target dari tujuan penelitian.

4.1 Simulasi Model Motor Induksi

Pada simulasi ini dilakukan pengujian dari perancangan model motor induksi yang telah dibahas pada bab sebelumnya yaitu perancangan sistem yang terdapat pada Gambar 3.1. Pada simulasi ini hanya dilihat respon kecepatan dan torsi. Pada pengujian ini motor induksi dioperasikan dengan beban dan tanpa beban. Respon kecepatan rotor dan torsi elektromagnetik pada motor induksi dengan beban dan tanpa beban dapat dilihat pada Gambar 4.1, 4.2, 4.3 dan 4.4.



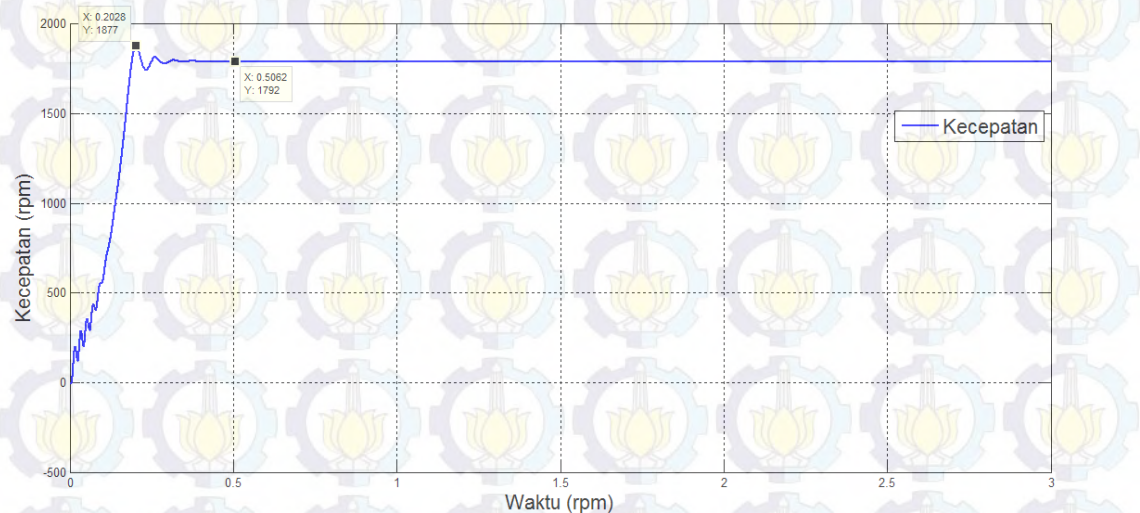
Gambar 4.1 Respon Kecepatan Rotor Model Motor Induksi Tanpa Beban

Gambar 4.1. menunjukkan respon putaran rotor motor induksi terdapat *overshoot* yang cukup besar ketika $t = 0.171$ detik, nilai maksimum *overshoot* yaitu 5.4111%. Dan ketika $t = 0.5$ detik, nilai kecepatan rotor 1800 rpm, pada saat inilah kecepatan rotor motor induksi mencapai *steady state*. Nilai kecepatan ini merupakan kondisi ketika motor induksi tanpa beban.



Gambar 4.2 ResponTorsi Elektromagnetik Model Motor Induksi Tanpa Beban

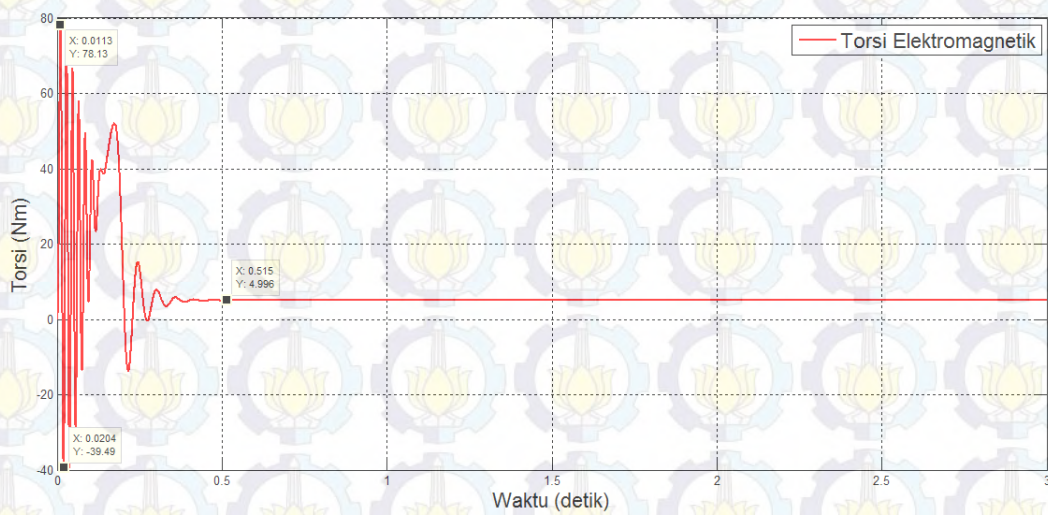
Respon torsi elektromagnetik yang terlihat pada Gambar 4.2. dapat kita lihat bahwa pada saat *start*, nilai *overshoot* yang dibangkitkan sangat besar yaitu -40.13 Nm sampai 77.74 Nm, akan tetapi nilai *overshoot* tersebut akan terus mengecil sampai $t = 0.5$ detik atau pada saat *steady state*, hal ini disebabkan karena nilai arus *starting* motor yang dibangkitkan besar. Gambar 4.3. merupakan respon putaran rotor motor induksi saat diberi beban.



Gambar 4.3 Respon Kecepatan Rotor Model Motor Induksi Berbeban

Respon kecepatan rotor pada motor induksi yang terlihat pada Gambar 4.3. dimana respon putaran rotor motor induksi terdapat masih terdapat *overshoot*

ketika $t = 0.2028$ detik, namun nilai maksimum *overshoot* lebih kecil jika dibandingkan sebelumnya. Ketika $t = 0.5$ detik, nilai kecepatan rotor mengalami penurunan dari yang sebelumnya (1800 rpm) yaitu 1792 rpm, pada saat inilah kecepatan rotor motor induksi mencapai *steady state*. hal ini dikarenakan adanya torsi beban, torsi beban yang diterapkan adalah sebesar 5 Nm. Gambar 4.4. adalah respon torsi motor induksi.



Gambar 4.4 Respon Torsi Elektromagnetik Model Motor Induksi Berbeban

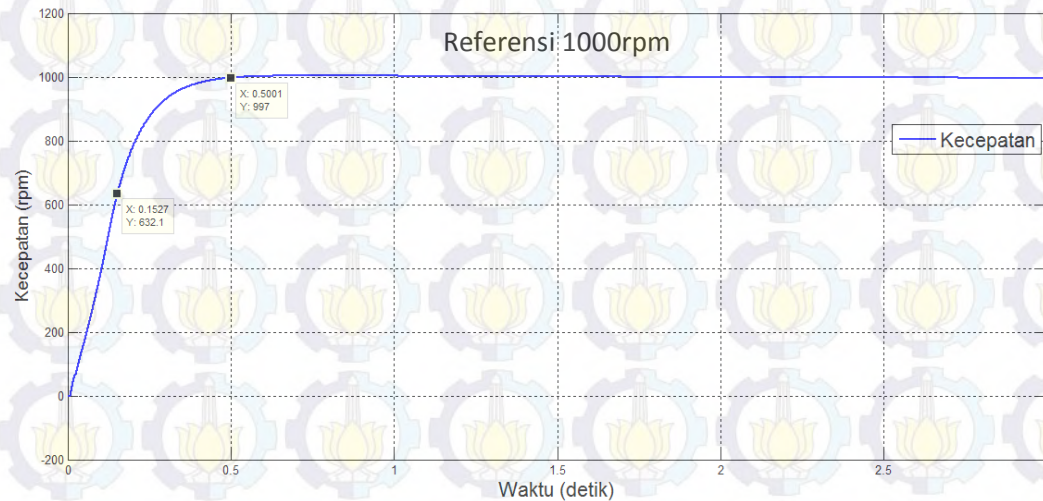
Respon torsi elektromagnetik yang terlihat pada Gambar 4.4. dapat kita lihat bahwa pada saat *start*, nilai *overshoot* yang dibangkitkan sebesar -39.49 Nm sampai 78.13 Nm, lebih besar dari yang sebelumnya, hal ini sesuai dengan teori karakteristik torsi motor. Nilai *overshoot* tersebut akan terus mengecil sampai $t = 0.5$ detik, di mana pada saat ini nilai torsi beban sebesar 5 Nm sesuai yang diterapkan. Pada saat waktu 0.5 detik, respon torsi mencapai *steady state*.

4.2 Simulasi Model Motor Induksi Dengan *Direct Torque Control* dan Kontroler Proporsional Integral (DTC-PI) Menggunakan *Fuzzy Logic Controller* (FLC) Berbeban

Simulasi yang dilakukan merupakan gabungan dari perancangan model motor induksi dengan DTC-PI Menggunakan metode *Fuzzy Logic Controller* (FLC). Torsi beban diberikan sebesar 5 Nm, dan nilai referensi untuk fluks stator ditentukan sebesar 1.46 Wb, sedangkan nilai acuan/referensi untuk kecepatan

sebesar 1000 rpm. Nilai referensi kecepatan inilah yang nantinya dijadikan acuan untuk nilai respon keluaran sistem yang diinginkan.

Hasil respon simulasi berupa respon kecepatan rotor, torsi elektromagnetik, fluk stator dan arus stator yang dapat dilihat pada Gambar 4.5, 4.6, 4.7, dan 4.8.



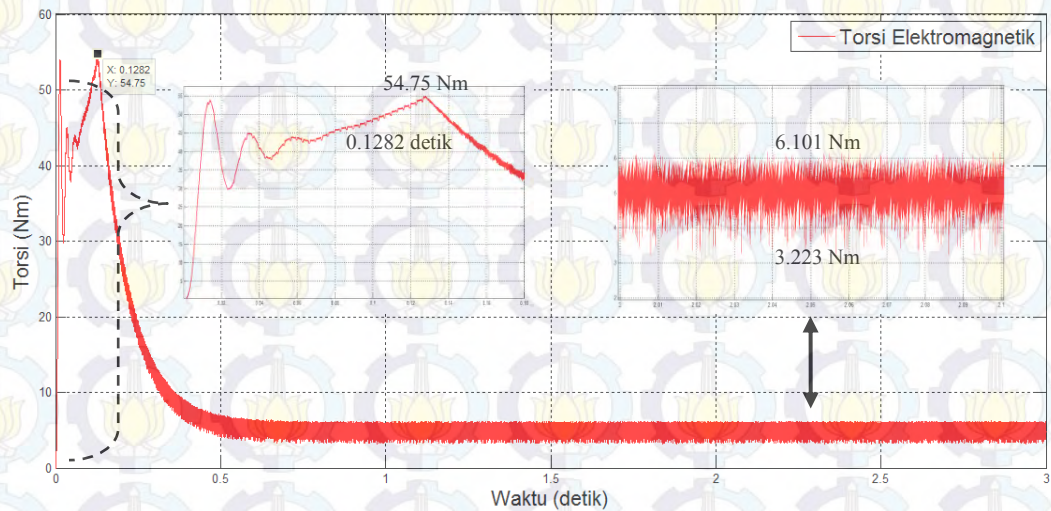
Gambar 4.5 Respon Kecepatan Rotor Motor Induksi dengan DTC-PI Menggunakan *Fuzzy Logic Controller* (FLC)

Pada Gambar 4.5 di atas sangat jelas sekali perbedaan antara respon kecepatan rotor model motor induksi (Gambar 4.1) dengan respon kecepatan rotor model motor induksi yang menggunakan DTC-PI dan *Fuzzy Logic Controller* (FLC). Pada respon kecepatan tersebut dapat dilihat bahwa tidak ada *overshoot* pada grafik respon, dan bentuk grafik respon seperti sistem orde pertama. Bentuk grafik respon yang seperti ini dikarenakan adanya penambahan kontroler PI-FLC pada sistem. Respon kecepatan mencapai nilai referensi yang ditentukan yaitu sekitar 1000 rpm. Nilai konstanta waktu (τ) dari sistem merupakan waktu ketika sistem mencapai 63,2 % dari *steady state* ($1000/\text{nilai referensi}$). Nilai konstanta waktu (τ) untuk sistem mencapai kondisi kecepatan $0,632 \times 1000 \text{ rpm} = 632 \text{ rad/menit}$. Sistem mencapai kondisi tersebut ketika $t = 0,1527 \text{ detik}$, sehingga nilai t inilah yang merupakan nilai konstanta waktu (τ) dari sistem. *Settling time* dapat diperoleh dengan menggunakan persamaan (4.1).

$$t_s(\pm 5\%) \approx 3\tau \quad (4.1)$$

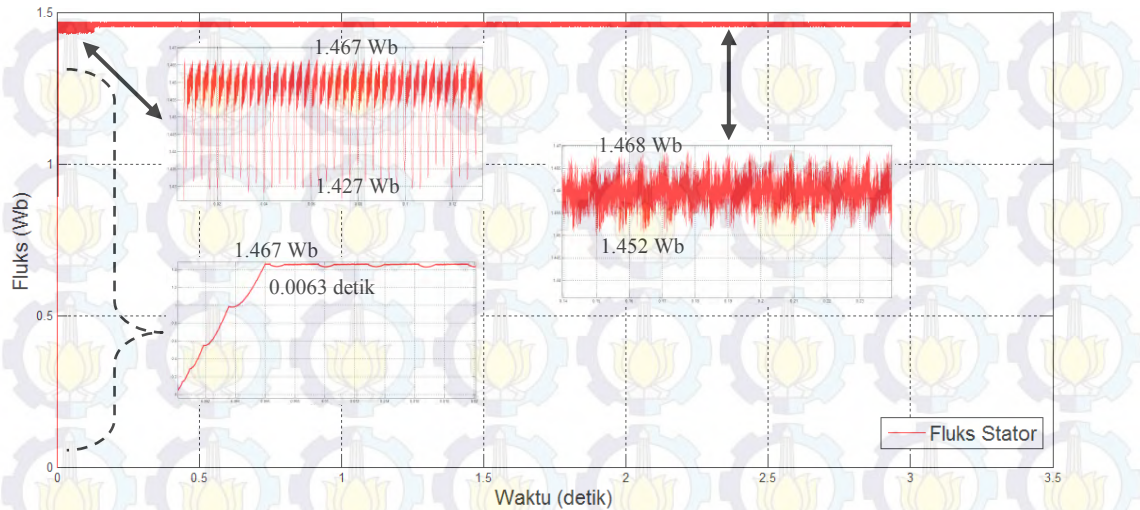
Sehingga *Settling time* yang diperoleh yaitu 0.4581 detik.

Respon torsi elektromagnetik yang diamati dapat dilihat pada Gambar 4.6.



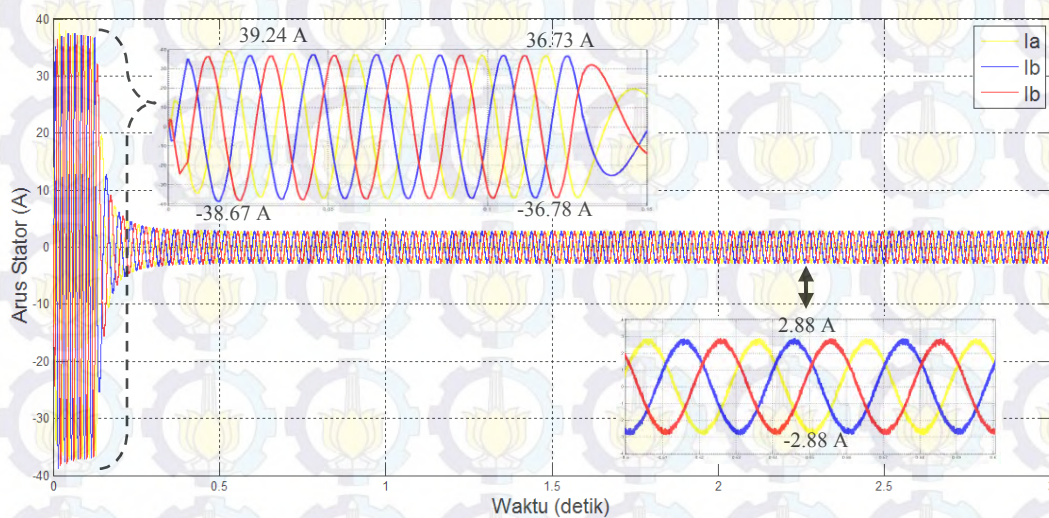
Gambar 4.6 Respon Torsi Elektromagnetik Motor Induksi dengan DTC-PI Menggunakan *Fuzzy Logic Controller* (FLC)

Penambahan blok kontroler proporsional integral (DTC-PI) menggunakan *Fuzzy Logic Controller* (FLC) pada model motor induksi sangat berpengaruh terhadap respon torsi elektromagnetik. Torsi beban diberikan sebesar 5 Nm. Respon awal torsi elektromagnetik pada saat *rise time* terjadi *overshoot* sebesar 54.75 Nm, yaitu pada saat waktu 0.1282 detik. Kemudian pada saat waktu 0.75 detik respon torsi elektromagnetik mencapai *steady state* dan pada saat *steady state*, fluktuasi *ripple* torsi berkisar antara 3.223 sampai dengan 6.101 Nm seperti yang terlihat pada Gambar 4.6 di atas respon torsi diperluas pada saat saat *steady state* yaitu dari 2 detik sampai 2.1 detik hal ini dilakukan untuk mempermudah dalam mengamati fluktuasi *ripple* torsi tersebut. Gambar 4.7. adalah Gambar respon fluks stator pada motor induksi dengan DTC-PI menggunakan *Fuzzy Logic Controller* (FLC).



Gambar 4.7 Respon Fluks Stator Motor Induksi dengan DTC-PI Menggunakan Fuzzy Logic Controller (FLC)

Jika diperhatikan Gambar 4.7 di atas, respon fluks stator pada motor induksi dengan DTC-PI menggunakan FLC, pada *steady state* pertama yaitu pada waktu 0.0063-0.1301 detik fluktuasi *ripple* fluks yang dihasilkan sangat besar yaitu berkisar antara 1.427-1.467 Wb. Ini menunjukkan bahwa dengan menggunakan DTC-PI-FLC pada motor induksi terjadi vibrasi pada saat kondisi *starting* awal. Kemudian pada saat waktu 0.1635 detik sampai dengan waktu seterusnya fluktuasi *ripple* fluks yang dihasilkan mengecil yaitu berkisar antara 1.452-1.468 Wb seperti yang ditunjukkan pada Gambar respon fluks diperluas saat *steady state* (0.5 detik sampai 0.6 detik). Respon arus dapat ditunjukkan pada Gambar 4.8.



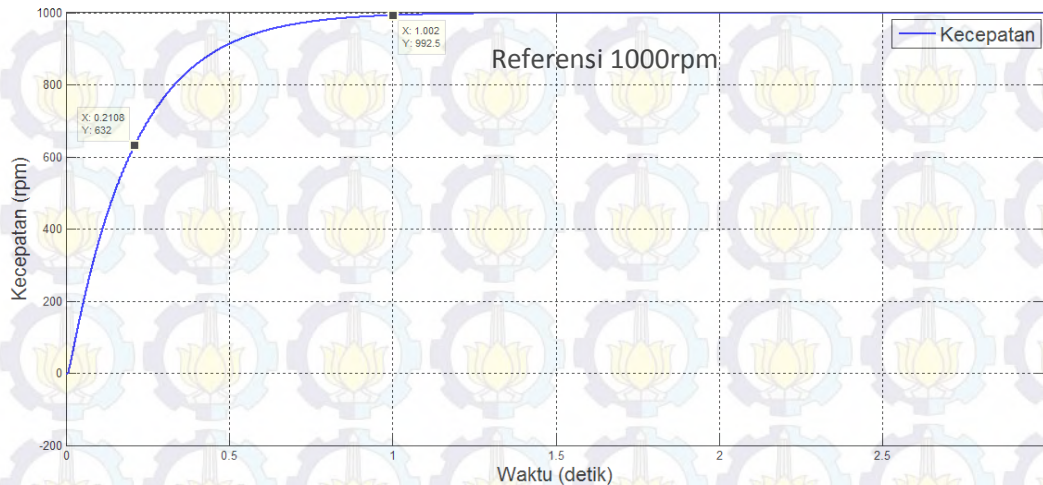
Gambar 4.8 Respon Arus Stator Motor Induksi dengan DTC-PI Menggunakan Fuzzy Logic Controller (FLC)

Berdasarkan Gambar 4.8. respon arus stator pada motor induksi di mana pada saat *rise time* yaitu pada saat waktu 0 sampai 0.14 detik, di mana pada saat itu terdapat *overshoot* yang sangat besar berkisar -38.67 sampai 39.24 Ampere. Hal ini menunjukkan bahwa pada *starting* awal arus stator yang dibangkitkan pada motor induksi sangat tinggi dibandingkan dengan arus pada saat *steady state* (arus nominal). Perbandingan *overshoot* dan nominal adalah arus *overshoot* dibagi arus nominal, di mana arus nominal bisa dilihat ketika respon arus diperluas pada saat *steady state* yaitu pada waktu 0.5 sampai 0.6 detik, pada saat ini nilai arus yaitu 2.88 Ampere, dan nilai 2.88 Ampere inilah sebagai arus nominal untuk motor induksi. Berarti $39.24 \text{ (arus overshoot)} / 2.88 \text{ (arus nominal)} = 13.625$. dengan demikian berarti arus *overshoot* 13 kali lebih besar dari arus nominal.

4.3 Simulasi Model Motor Induksi Dengan *Direct Torque Control* dan Kontroler Proporsional Integral (DTC-PI) Menggunakan *Sliding Mode Control* (SMC) Berbeban

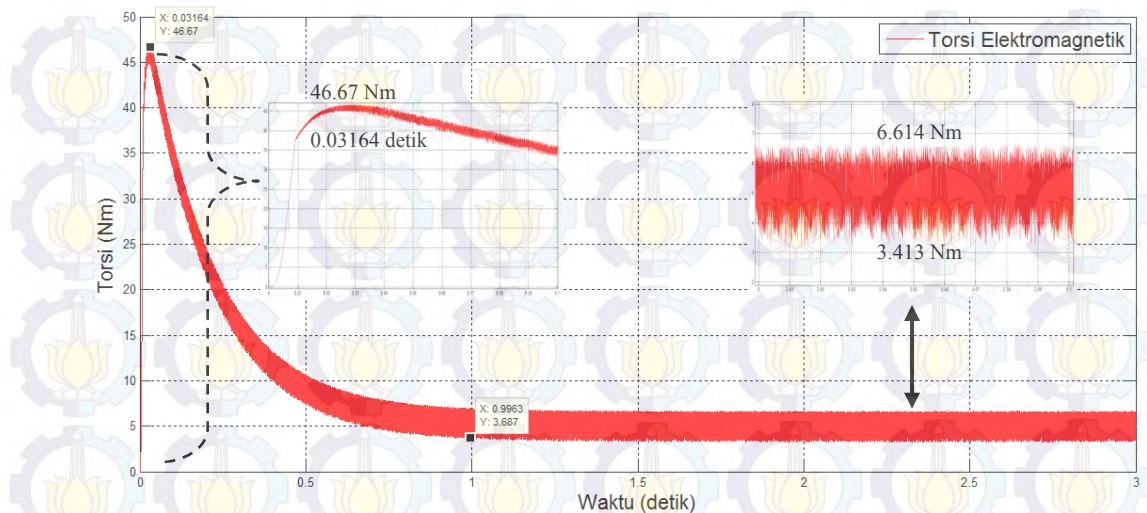
Simulasi yang dilakukan merupakan gabungan dari perancangan model motor induksi dengan DTC-PI Menggunakan metode *Sliding Mode Control* (SMC). Pada percobaan ini torsi beban diberikan sebesar 5 Nm, dan nilai referensi untuk fluks stator ditentukan sebesar 1.46 Wb, sedangkan nilai acuan/referensi untuk kecepatan sebesar 1000 rpm. Nilai referensi kecepatan inilah yang nantinya dijadikan acuan untuk nilai respon keluaran sistem yang diinginkan.

Hasil respon simulasi berupa respon kecepatan rotor, torsi elektromagnetik, fluks stator dan arus stator yang dapat dilihat pada Gambar 4.9, 4.10, 4.11, dan 4.12.



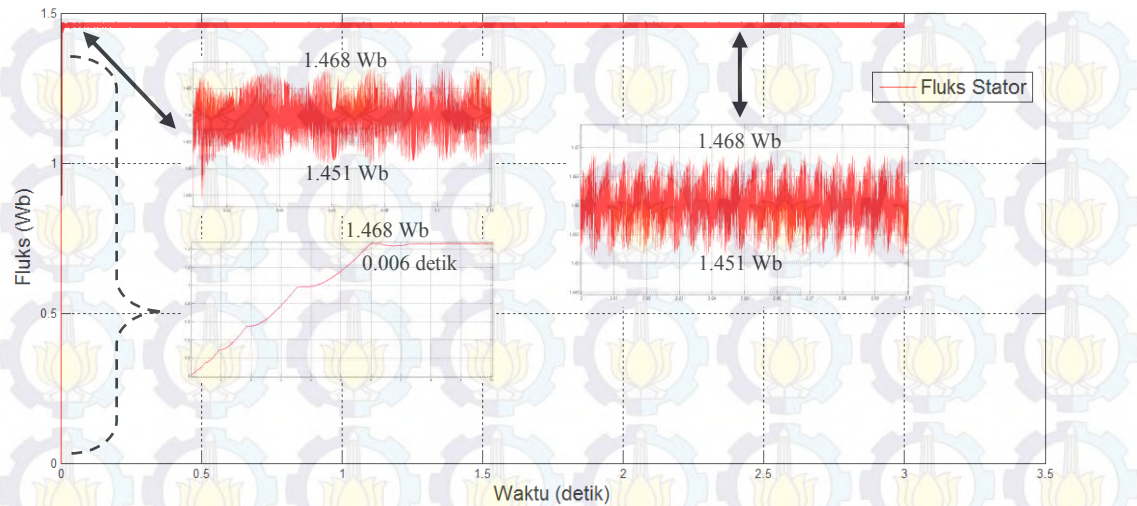
Gambar 4.9 Respon Kecepatan Rotor Motor Induksi dengan DTC-PI Menggunakan *Sliding Mode Control* (SMC)

Berdasarkan Gambar 4.9 di atas respon kecepatan motor induksi dengan DTC-PI menggunakan SMC dapat dilihat bahwa tidak ada *overshoot* pada grafik respon, dan bentuk grafik respon seperti sistem orde pertama. Bentuk grafik respon yang seperti ini dikarenakan adanya penambahan kontroler PI-SMC pada sistem. Respon kecepatan mencapai nilai referensi yang ditentukan yaitu 1000 rpm. Nilai konstanta waktu (τ) dari sistem merupakan waktu ketika sistem mencapai 63,2 % dari *steady state* ($1000/\text{nilai referensi}$). Nilai konstanta waktu (τ) untuk sistem mencapai kondisi kecepatan $0,632 \times 1000 \text{ rpm} = 632 \text{ rpm}$. Sistem mencapai kondisi tersebut ketika $t = 0,2108$ detik, sehingga nilai t inilah yang merupakan nilai konstanta waktu (τ) dari system. Dengan demikian berdasarkan grafik respon di atas dan dengan menggunakan persamaan (4.1) diperoleh *Settling time* sebesar 0.6324 detik. Gambar 4.10. adalah Gambar respon torsi elektromagnetik.



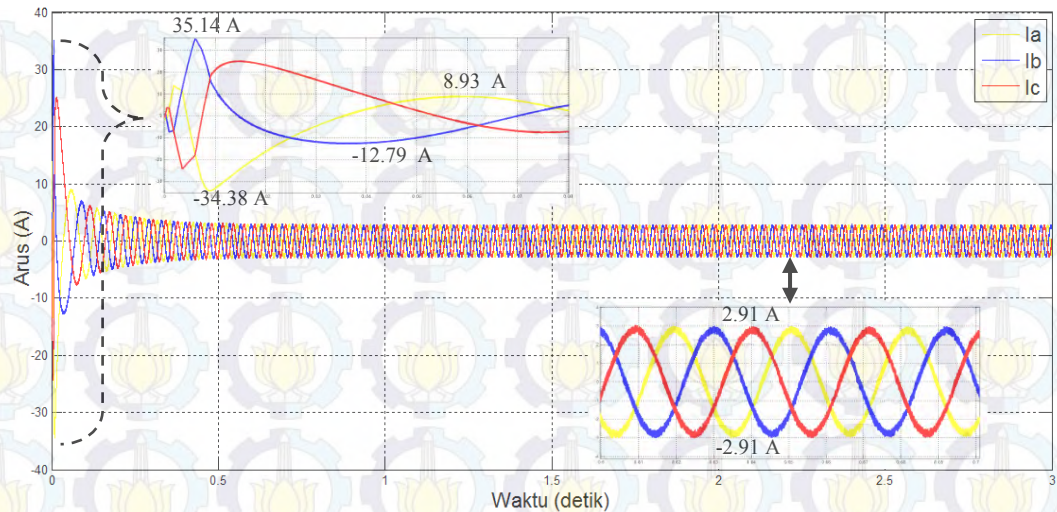
Gambar 4.10 Respon Torsi Elektromagnetik Motor Induksi dengan DTC-PI Menggunakan *Sliding Mode Control* (SMC)

Jika kita amati Gambar 4.10 di atas, terlihat jelas perbedaan antara respon torsi elektromagnetik (DTC-PI-FLC) dengan respon torsi elektromagnetik (DTC-PI-SMC) pada model motor induksi. Jika sebelumnya (DTC-PI-FLC), respon awal torsi elektromagnetik pada saat *rise time* terjadi *overshoot* sebesar 54.75 Nm, dapat diminimalisasi dengan menggunakan DTC-PI-SMC yaitu *overshoot* yang terjadi menjadi 46.67 Nm, dan pada saat waktu 0.03164 detik. Pada saat ini torsi beban diberikan sama dengan yang sebelumnya yaitu sebesar 5 Nm. Kemudian pada saat waktu 0.99 detik respon torsi elektromagnetik mencapai *steady state* dan pada saat *steady state*, fluktuasi *ripple* torsi berkisar antara 3.413 sampai dengan 6.614 Nm seperti yang terlihat pada Gambar 4.10 di atas respon torsi diperluas pada saat saat *steady state* yaitu dari 2 detik sampai 2.1 detik hal ini dilakukan untuk mempermudah dalam mengamati fluktuasi *ripple* torsi tersebut. Respon fluks stator dapat dilihat pada Gambar 4.11.



Gambar 4.11 Respon Fluks Stator Motor Induksi dengan DTC-PI Menggunakan *Sliding Mode Control* (SMC)

Dari Gambar 4.11 di atas, respon fluks stator pada motor induksi dengan DTC-PI menggunakan SMC, pada *steady state* pertama sampai dengan seterusnya fluktuasi *ripple* fluks yang dihasilkan berkisar antara 1.451-1.468 Wb. Hal ini menunjukkan bahwa terdapat perbedaan dengan yang sebelumnya yaitu respon fluks stator dengan DTC-PI-FLC pada motor induksi. Fluktuasi *ripple* fluks yang dibangkitkan pada motor induksi dengan DTC-PI menggunakan SMC pada saat *steady state* awal dan seterusnya adalah sekitar 0.017 Wb, nilai ini didapatkan dengan cara nilai fluks tertinggi dikurang nilai fluks terendah (nilai fluks diantara referensi). Berikut adalah respon arus stator motor induksi yang ditunjukkan pada Gambar 4.12.



Gambar 4.12 Respon Arus Stator Motor Induksi dengan DTC-PI Menggunakan *Sliding Mode Control (SMC)*

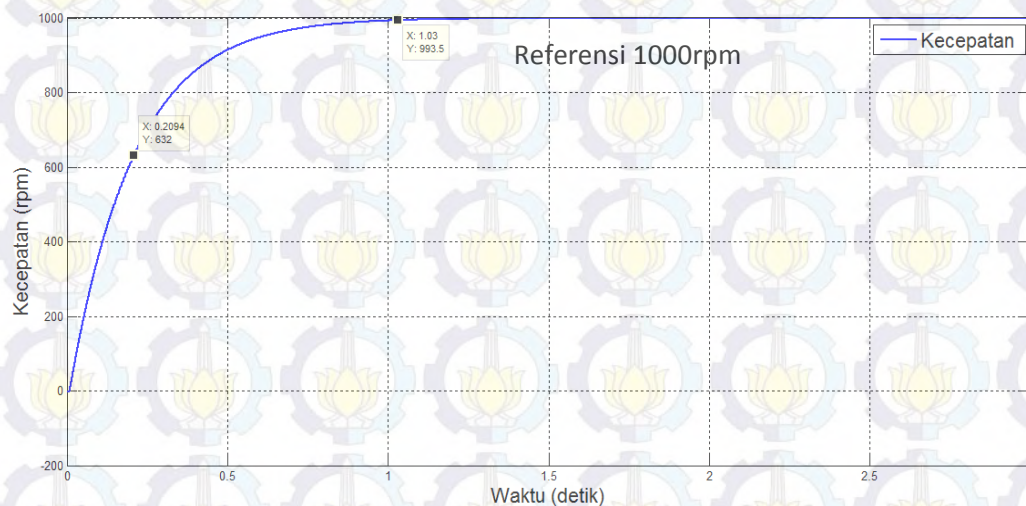
Gambar 4.12. menunjukkan bahwa respon arus stator pada motor induksi di mana pada saat *rise time* yaitu pada saat waktu 0 sampai 0.08 detik, di mana pada saat itu terjadi *overshoot* sebesar -34.38 Ampere ini terjadi pada saat waktu 0.00924 detik dan *overshoot* sebesar 35.14 Ampere ini terjadi ketika waktu 0.00606 detik. Kemudian pada waktu 0.04 detik *overshoot* menurun yaitu sebesar -12.79 Ampere dan pada waktu 0.06 detik sebesar 8.93 Ampere. Hal ini membuktikan bahwa *overshoot* yang terjadi lebih kecil dan lebih pendek durasinya dibandingkan dengan yang sebelumnya (respon arus dengan DTC-PI-FLC). Pada saat *steady state* arus yang dibangkitkan sebesar 2.91 Ampere, nilai arus inilah sebagai nilai nominal, dengan demikian dapat kita tentukan bahwa $35.14 \text{ Ampere (arus overshoot)} / 2.91 \text{ Ampere (arus nominal)} = 12.075$. Dengan demikian bahwa *overshoot* yang terjadi kurang lebih 12 kali lebih besar dari arus nominal atau arus pada saat *steady state*.

4.4 Simulasi Model Motor Induksi Dengan *Direct Torque Control (DTC)* Menggunakan *Sliding Mode Control (SMC)* Berbasis *Algoritma Genetika* Berbeban

Simulasi ini merupakan gabungan dari perancangan model motor induksi dengan DTC-SMC Berbasis *Algoritma Genetika* (AG). Pada simulasi ini torsi beban diberikan sebesar 5 Nm, dan nilai referensi untuk fluks stator ditentukan

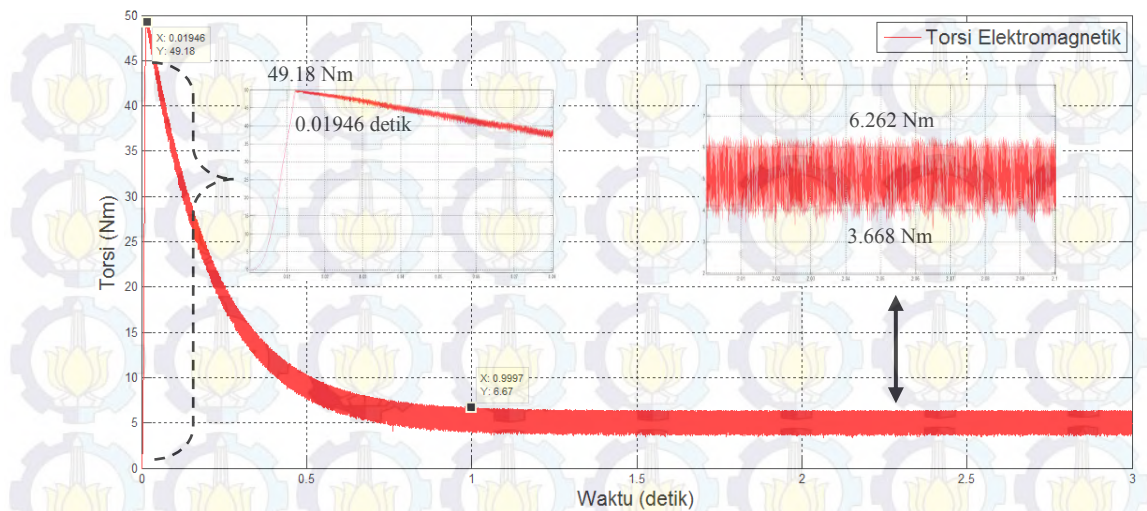
sebesar 1.46 Wb, sedangkan nilai acuan/referensi untuk kecepatan sebesar 1000 rpm. Nilai referensi kecepatan inilah yang nantinya dijadikan acuan untuk nilai respon keluaran sistem yang diinginkan.

Hasil respon simulasi berupa respon kecepatan rotor, torsi elektromagnetik, fluk stator dan arus stator yang dapat dilihat pada Gambar 4.13, 4.14, 4.15, dan 4.16.



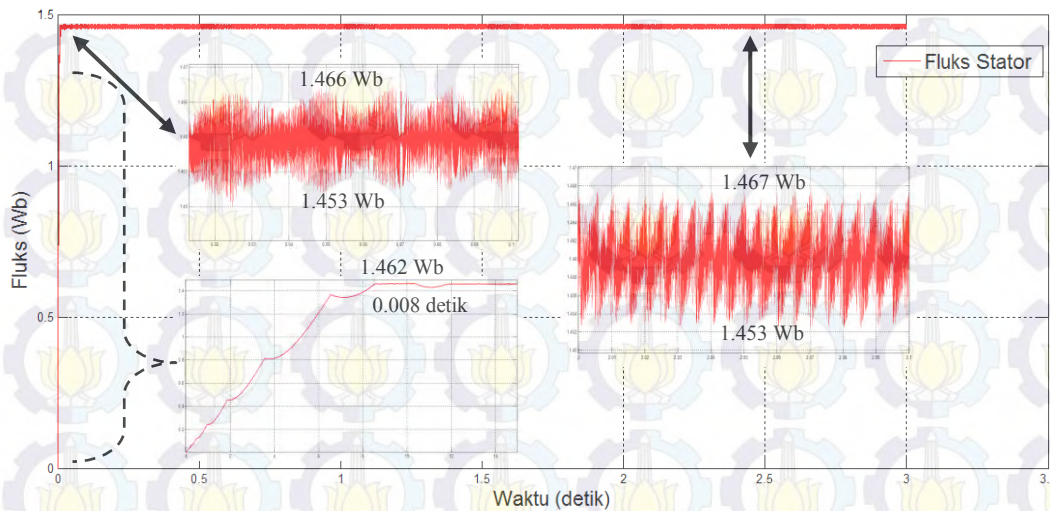
Gambar 4.13 Respon Kecepatan Rotor Motor Induksi DTC Menggunakan *Sliding Mode Control (SMC)* Berbasis *Algoritma Genetika*

Berdasarkan Gambar 4.13 di atas respon kecepatan motor induksi dengan DTC-SMC berbasis *Algoritma Genetika* dapat dilihat bahwa tidak ada *overshoot* pada grafik respon, dan bentuk grafik respon seperti sistem orde pertama. Bentuk grafik respon yang seperti ini dikarenakan adanya penambahan kontroler SMC pada sistem. Respon kecepatan mencapai nilai referensi yang ditentukan yaitu 1000 rpm. Nilai konstanta waktu (τ) dari sistem merupakan waktu ketika sistem mencapai 63.2 % dari *steady state* (1000/nilai referensi). Nilai konstanta waktu (τ) untuk sistem mencapai kondisi kecepatan $0.632 \times 1000 \text{ rpm} = 632 \text{ rpm}$. Sistem mencapai kondisi tersebut ketika $t = 0.2094$ detik, sehingga nilai t inilah yang merupakan nilai konstanta waktu (τ) dari system. Dengan demikian berdasarkan grafik respon di atas dan dengan menggunakan persamaan (4.1) diperoleh *Settling time* sebesar 0.6282 detik. Respon torsi elektromagnetik dapat ditunjukkan pada Gambar 4.14.



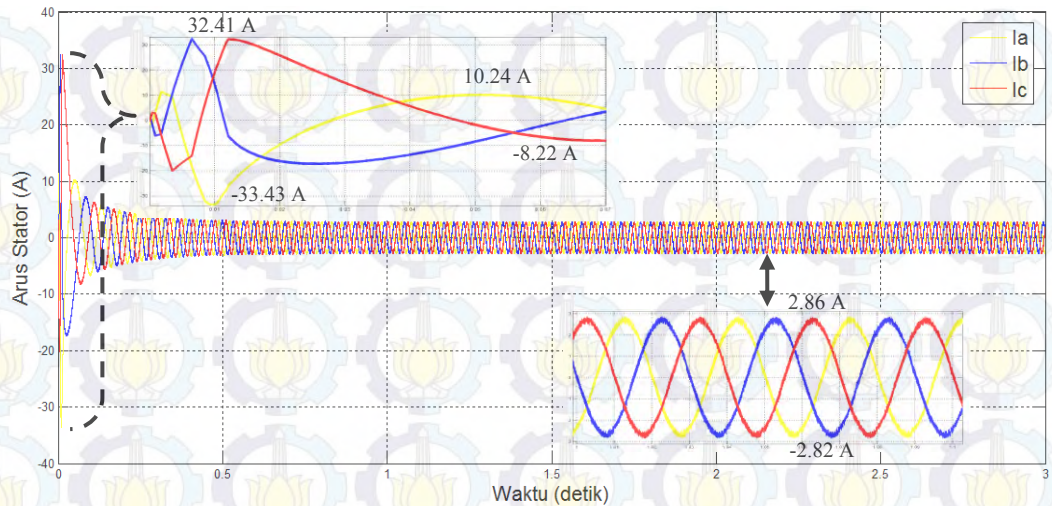
Gambar 4.14 Respon Torsi Elektromagnetik Motor Induksi dengan DTC Menggunakan *Sliding Mode Control* (SMC) Berbasis *Algoritma Genetika*

Jika kita amati Gambar 4.14 di atas, hampir tidak ada perbedaan antara respon torsi elektromagnetik (DTC-PI-SMC) dengan respon torsi elektromagnetik (DTC-SMC-AG) pada motor induksi. Namun dengan digunakan metode DTC-SMC berbasis *Algoritma Genetika*, fluktuasi *ripple* torsi yang dibangkitkan pada saat *steady state* sebesar 3.668 Nm sampai dengan 6.262 Nm. Hal ini membuktikan bahwa fluktuasi *ripple* torsi lebih kecil jika dibandingkan dengan sebelumnya (DTC-PI-SMC). Dengan kata lain bahwa fluktuasi *ripple* torsi yang dibangkitkan pada saat *steady state* sebesar 2.594 Nm jika sebelumnya 3.201 Nm. Nilai ini didapatkan dengan cara menghitung jarak antara nilai *ripple* torsi terendah ke nilai *ripple* torsi tertinggi. Dan respon torsi pada saat *rise time* terdapat *overshoot* sebesar 49.18 Nm pada waktu 0.01946 detik. Respon fluks stator dapat dilihat pada Gambar 4.15.



Gambar 4.15 Respon Fluks Stator Motor Induksi dengan DTC Menggunakan *Sliding Mode Control* (SMC) Berbasis *Algoritma Genetika*

Dari Gambar 4.15 di atas, respon fluks stator pada motor induksi dengan DTC-SMC berbasis *Algoritma Genetika*, pada *steady state* pertama sampai dengan seterusnya fluktuasi *ripple* fluks yang dihasilkan berkisar antara 1.453-1.466 Wb. Hal Ini membuktikan bahwa fluktuasi *ripple* fluks dapat diminimalisasi dengan adanya optimalisasi parameter SMC dengan menggunakan *Algoritma Genetika*. Fluktuasi *ripple* fluks yang dibangkitkan adalah sebesar 0.013 Wb, di mana sebelumnya (DTC-SMC) fluktuasi *ripple* fluks yang dibangkitkan adalah sebesar 0.017 Wb. Nilai ini didapatkan dengan cara nilai fluks tertinggi dikurang nilai fluks terendah (nilai fluks di antara referensi). Berikut adalah respon arus stator motor induksi yang ditunjukkan pada Gambar 4.16.

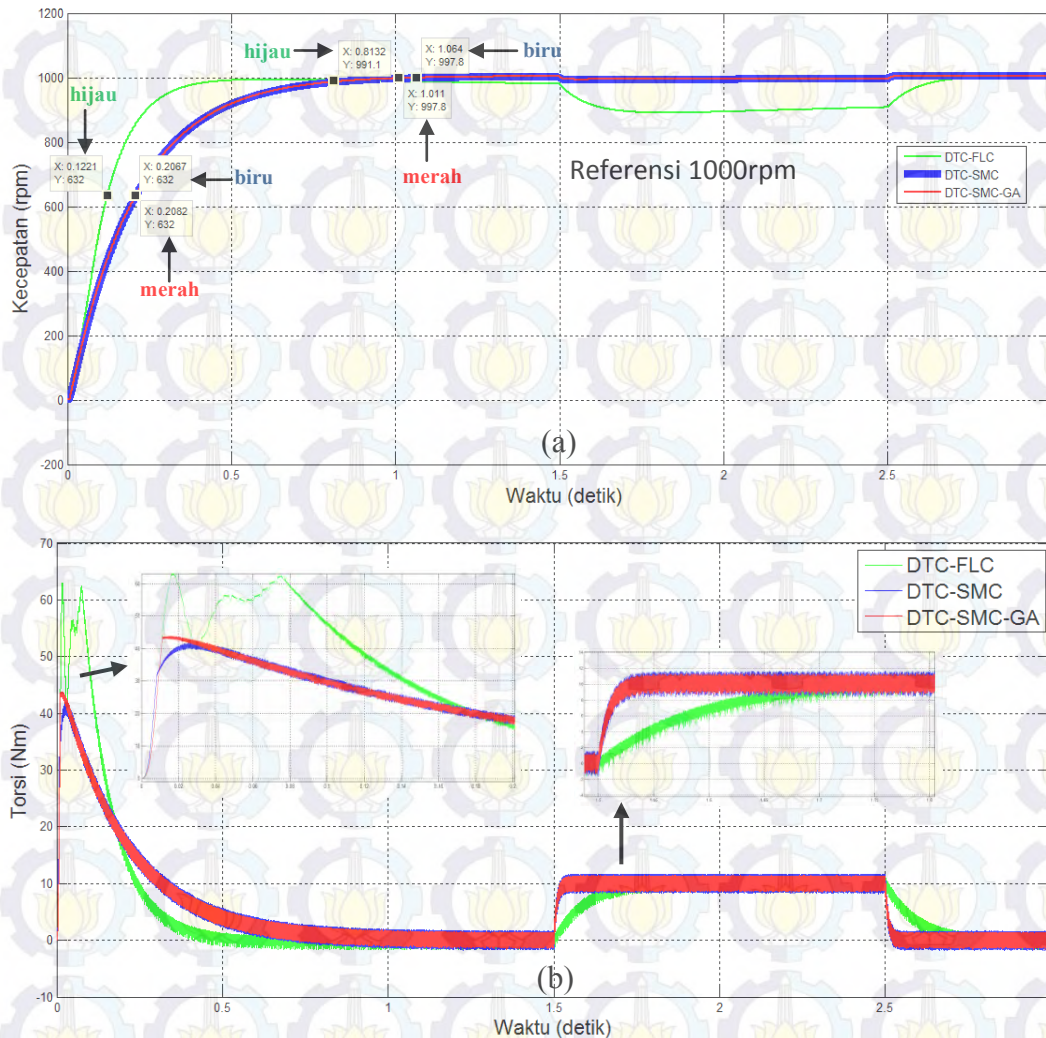


Gambar 4.16 Respon Arus Stator Motor Induksi dengan DTC Menggunakan *Sliding Mode Control (SMC)* Berbasis *Algoritma Genetika*

Dari Gambar 4.16 menunjukkan bahwa respon arus stator pada motor induksi di mana pada saat *rise time* terjadi *overshoot* sebesar -33.43 Ampere sampai dengan 32.41 Ampere. Kemudian *overshoot* semakin mengecil sampai dengan pada saat *steady state*. *Overshoot* yang terjadi nilainya lebih kecil jika dibandingkan dengan sebelumnya. Penggunaan metode DTC-SMC berbasis *Algoritma Genetika* membuktikan bahwa *overshoot* yang terjadi kurang lebih 11 kali lebih besar dari arus nominal atau arus pada saat *steady state*. Arus nominal atau pada saat *steady state* adalah sebesar 2.86 Ampere.

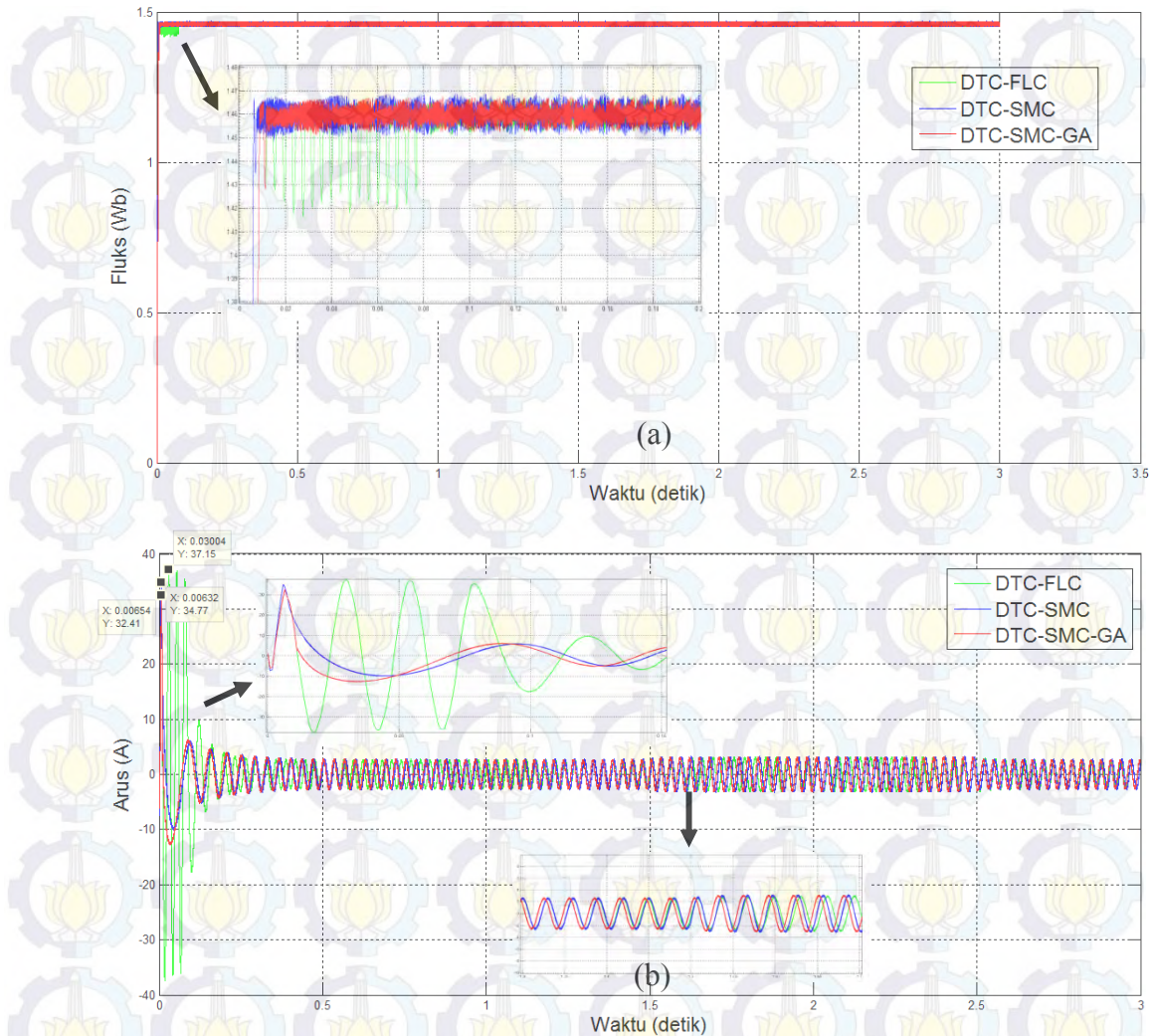
4.5 Simulasi Perbandingan Respon Kecepatan Motor Induksi Tiga Fasa Dengan Metode Kontroller Berbeda

Perbandingan respon kecepatan motor induksi dilakukan untuk mengetahui kemampuan sistem. Pada simulasi ini kita ingin melihat respon putaran motor terhadap perubahan torsi beban, dengan nilai acuan/referensi untuk kecepatan berbeda-beda. Nilai referensi kecepatan inilah yang nantinya dijadikan acuan untuk nilai respon keluaran sistem yang diinginkan. Gambar 4.15 menunjukkan respon kecepatan dan torsi motor induksi.



Gambar 4.17 (a) Respon Kecepatan (b) Respon Torsi (Motor Induksi dengan DTC-FLC, DTC-SMC dan DTC-SMC-GA, dengan kecepatan referensi 1000 rpm).

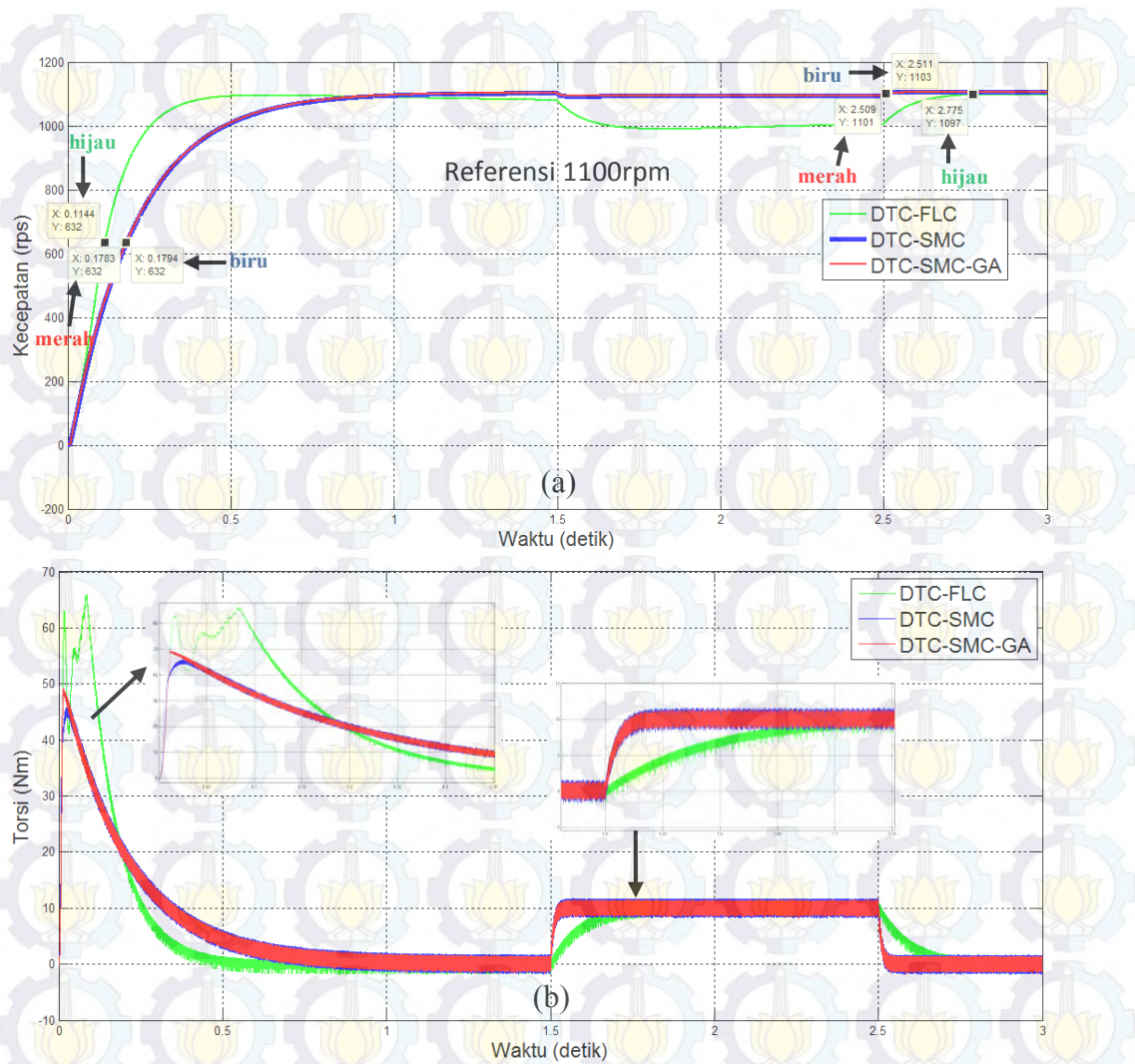
Gambar 4.17 menunjukkan respon kecepatan dan torsi motor induksi ketika diberi beban sebesar 10 Nm, diterapkan pada 1.5 detik dan beban dihilangkan pada 2.5 detik dengan menggunakan kontroler yang berbeda yaitu DTC-FLC, DTC-SMC dan DTC-SMC-GA dengan referensi kecepatan 1000 rpm. Respon kecepatan dengan DTC-SMC dan DTC-SMC-GA terlihat bahwa tidak ada perbedaan yang signifikan dibandingkan dengan DTC-FLC, namun pada respon torsi jika diamati terdapat perbedaan yaitu fluktuasi *ripple* minimum. Dan untuk respon fluk stator dan arus stator dapat dilihat pada Gambar 4.18.



Gambar 4.18 (a) Respon Fluks Stator (b) Respon Arus Stator (Motor Induksi dengan DTC-FLC, DTC-SMC dan DTC-SMC-GA. dengan kecepatan referensi 1000rpm).

Gambar 4.18 menunjukkan respon fluks stator dan arus stator pada motor induksi dengan kontroler berbeda, di mana pada DTC-FLC pada saat *steady state* awal fluktuasi *ripple* fluks yang dibangkitkan sangat besar, begitu juga dengan arus *starting* awal yang dibangkitkan sekitar 37.15A dan waktu *steady state* lebih lama dibandingkan dengan menggunakan kontroler DTC-SMC dan DTC-SMC-GA.

Selanjutnya sistem diberi masukan referensi yang berbeda, yaitu di atas 1000 rpm. Hal ini dilakukan untuk mengetahui apakah ada perbedaan karakteristik respon sistem saat diberi masukan kecepatan referensi yang berbeda-beda. Gambar 4.19 merupakan respon kecepatan dan torsi.



Gambar 4.19 (a) Respon Kecepatan (b) Respon Torsi (Motor Induksi dengan DTC-FLC, DTC-SMC dan DTC-SMC-GA, dengan kecepatan referensi 1100rpm)

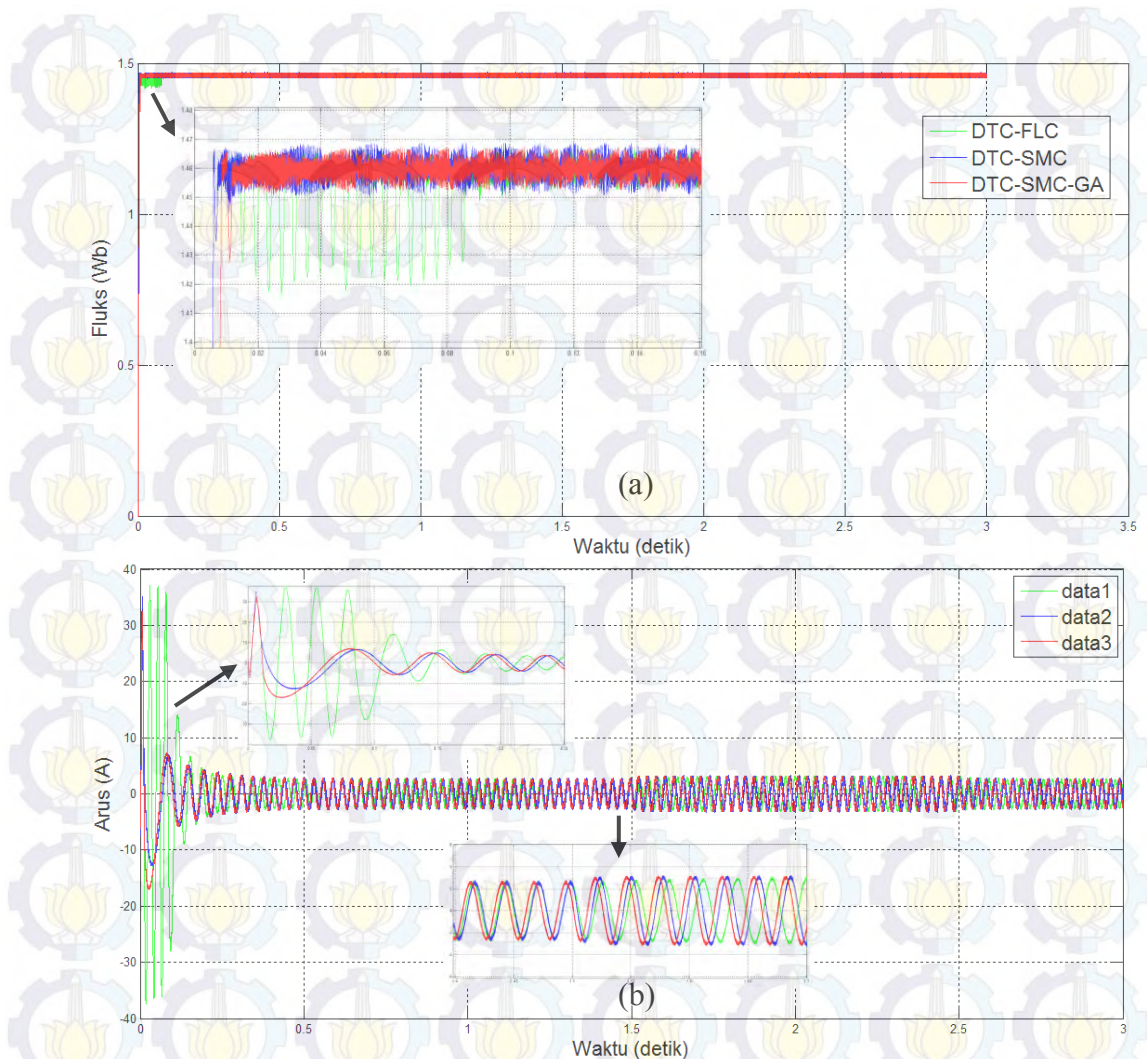
Gambar 4.19 menunjukkan respon kecepatan dan torsi motor induksi dengan referensi kecepatan 1100 rpm. Ketika beban diterapkan yaitu pada waktu 1.5 detik, respon kecepatan DTC-SMC dan DTC-SMC-GA terlihat hampir tidak mengalami perubahan kecepatan dari referensi dibandingkan dengan DTC-FLC dan waktu respon torsi mencapai *steady state* lebih cepat yaitu 1.55 detik. Gambar 4.19 menunjukkan karakteristik respon sistem sebagai berikut (DTC-SMC-GA):

$$\tau = 0.1783 \text{ detik}$$

$$t_s(5\%) = 3\tau = 0.5349$$

$$t_r(10\% - 90\%) = \tau \ln 9 = 0.3917$$

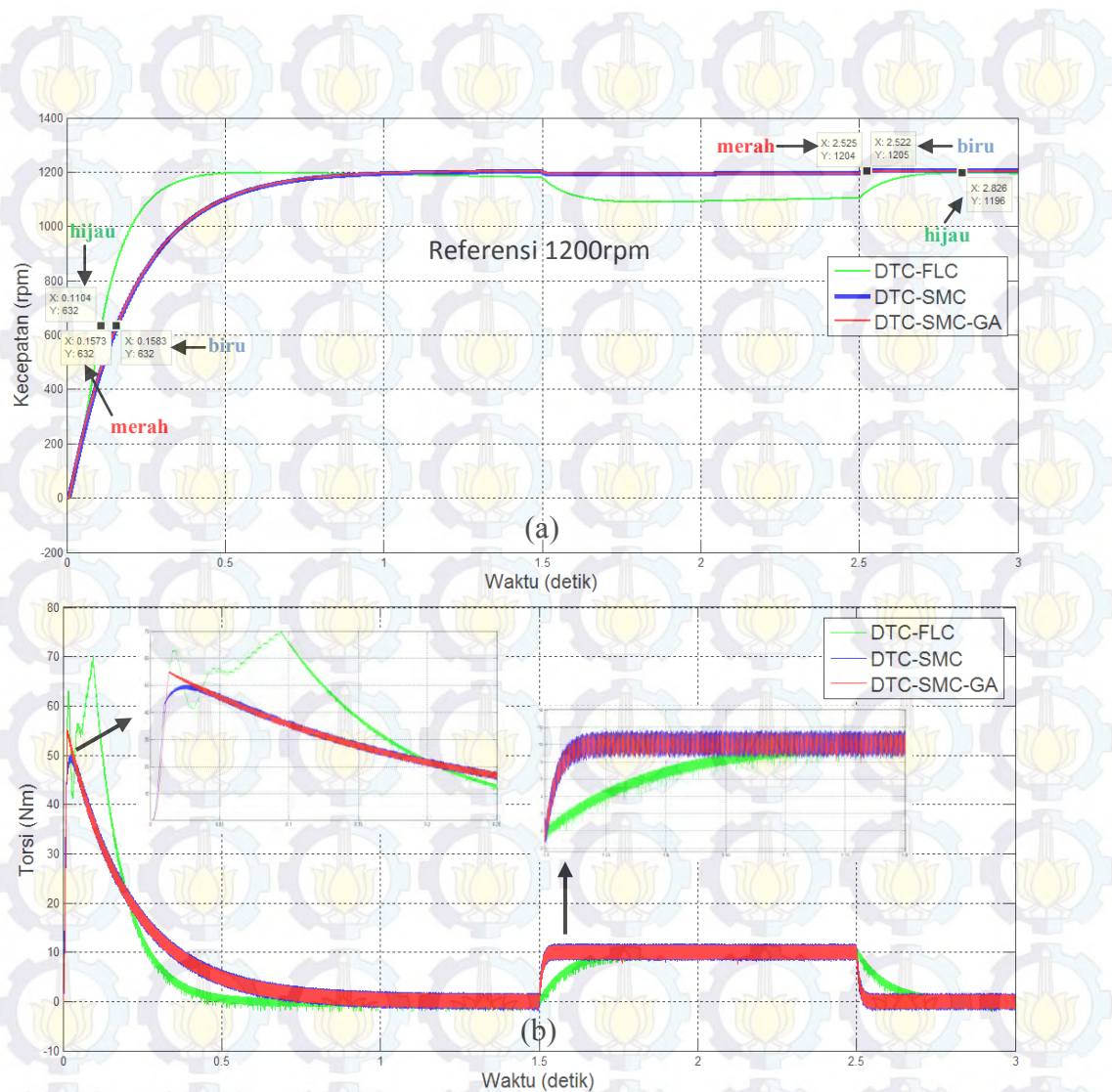
$$t_d = \tau \ln 2 = 0.1235$$



Gambar 4.20 (a) Respon Fluks Stator (b) Respon Arus Stator (Motor Induksi dengan DTC-FLC, DTC-SMC dan DTC-SMC-GA, dengan kecepatan referensi 1100rpm)

Gambar 4.20 menunjukkan respon fluks stator dan arus stator pada motor induksi dengan kontroler berbeda, di mana pada DTC-FLC pada saat *steady state* awal fluktuasi *ripple* fluks yang dibangkitkan besar namun pada *steady state* seterusnya fluktuasi *ripple* fluks mengecil dan pada saat *starting* awal respon arus terdapat *overshoot* yang sangat besar, waktu mencapai *steady state* lebih lama jika dibandingkan dengan menggunakan kontroler DTC-SMC dan DTC-SMC-GA.

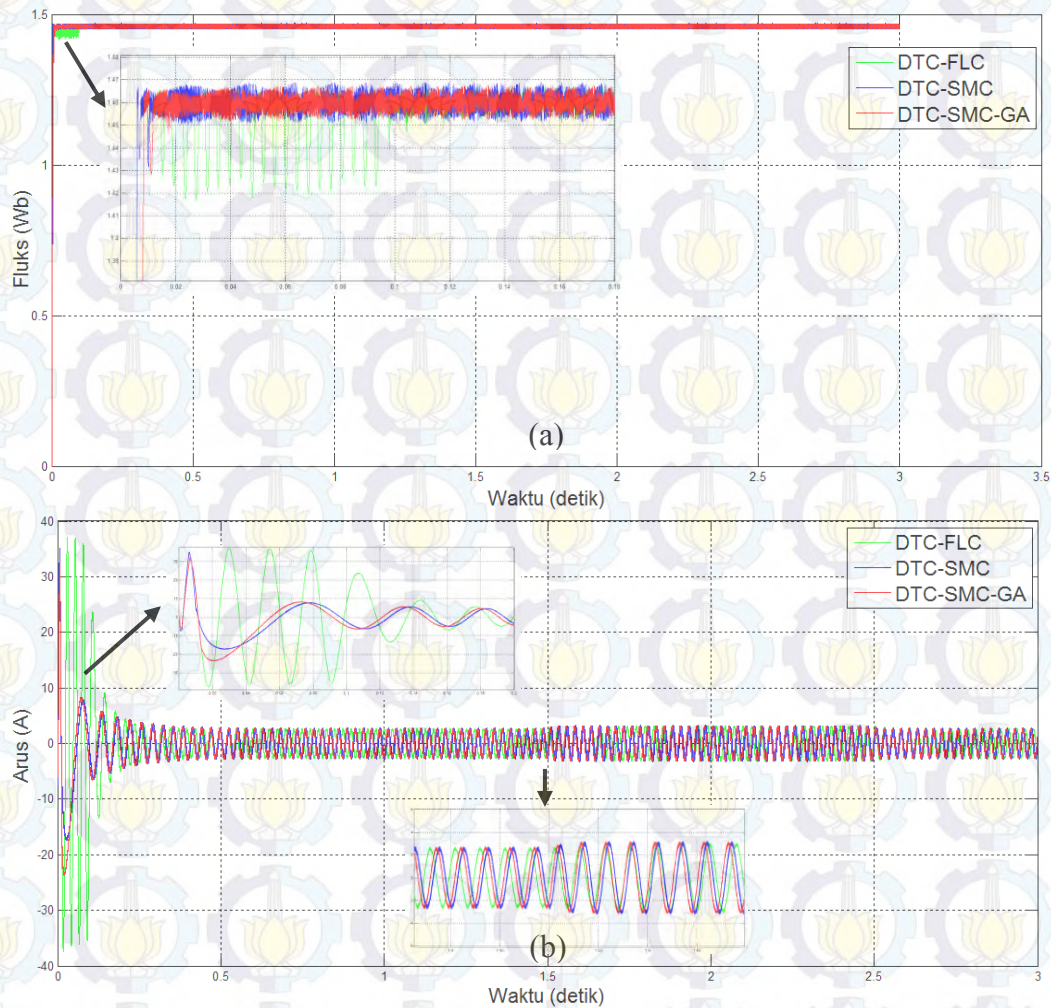
Pada pengujian selanjutnya sistem diberi masukan referensi yang berbeda, yaitu kecepatan referensi sebesar 1200 rpm. Hal ini juga dilakukan untuk mengetahui apakah ada perbedaan karakteristik respon sistem. Respon kecepatan dan torsi saat diberi referensi sebesar 1200 rpm dapat dilihat dari Gambar 4. 21.



Gambar 4.21 (a) Respon Kecepatan (b) Respon Torsi (Motor Induksi dengan DTC-FLC, DTC-SMC dan DTC-SMC-GA, dengan kecepatan referensi 1200rpm)

Gambar 4.21 menunjukkan respon kecepatan dan torsi motor induksi dengan referensi kecepatan 1200 rpm dengan kontroler yang berbeda. Respon kecepatan dengan DTC-FLC waktu *steady state* lebih cepat yaitu 0.5036 detik dibandingkan dengan DTC-SMC dan DTC-SMC-GA yaitu sekitar 0.9966 detik, namun pada saat diberi beban respon kecepatan dengan DTC-FLC mengalami penurunan kecepatan yang sangat signifikan dari referensi yang diberikan dan pada saat beban dilepas (waktu 2.5 detik) respon kecepatan dengan DTC-FLC waktu mencapai *steady state* lebih lama yaitu 2.826 detik, sementara pada DTC-

SMC waktu mencapai *steady state* pada saat itu adalah 2.522 detik. Hal ini membuktikan bahwa metode SMC mampu mengatasi perubahan beban.



Gambar 4.22 (a) Respon Fluks Stator (b) Respon Arus Stator (Motor Induksi dengan DTC-FLC, DTC-SMC dan DTC-SMC-GA, dengan kecepatan referensi 1200rpm).

Gambar 4.22 menunjukkan respon fluks stator dan arus stator pada motor induksi dengan kontroler berbeda. Perbedaan terlihat pada saat *starting* awal motor induksi. Hal ini sama dengan pada pengujian sebelumnya yaitu dengan kecepatan referensi 1000 rpm dan 1100 rpm. Respon arus pada saat torsi beban diberikan, yaitu pada waktu 1.5 detik, dimana pada kondisi ini adalah kondisi *steady state* terlihat bahwa arus mengalami kenaikan sebesar 3.049A. (DTC-SMC-GA) di mana sebelumnya adalah 2.63A. hal ini sesuai dengan teori karakteristik torsi motor.

Dari tiga kali pengujian dengan referensi kecepatan berbeda, hasilnya menunjukkan adanya perbedaan dari karakteristik masing-masing sistem. Tabel 4.1 menunjukkan karakteristik respon waktu masing-masing sistem.

Tabel 4.1 Perbandingan Karakteristik Respon Waktu (DTC-FLC, DTC-SMC dan DTC-SMC-GA)

| Kecepatan (rpm) | Karakteristik respon | DTC-FLC (detik) | DTC-SMC (detik) | DTC-SMC-GA (detik) |
|--------------------|-------------------------|--------------------|--------------------|-----------------------|
| 1000 | τ | 0,1221 | 0,2067 | 0,2082 |
| | t_s | 0,3663 | 0,6201 | 0,6246 |
| | t_r | 0,2682 | 0,4541 | 0,4574 |
| | t_d | 0,0846 | 0,1432 | 0,1443 |
| 1100 | τ | 0,1144 | 0,1794 | 0,1783 |
| | t_s | 0,3432 | 0,5382 | 0,5349 |
| | t_r | 0,2513 | 0,3941 | 0,3917 |
| | t_d | 0,0792 | 0,1243 | 0,1235 |
| 1200 | τ | 0,1104 | 0,1583 | 0,1753 |
| | t_s | 0,3312 | 0,4749 | 0,5259 |
| | t_r | 0,2425 | 0,3478 | 0,3851 |
| | t_d | 0,0765 | 0,1096 | 0,2669 |

Dari Tabel 4.1 di atas dapat dilihat adanya perbedaan karakteristik respon sistem dengan kontroler yang berbeda. Namun untuk sistem kecepatan yang berbeda, perbedaan terjadi pada waktu *rise time*. Dengan digunakan kontroler DTC-FLC, nilai konstanta waktu (τ) dengan referensi yang berbeda rata-rata adalah 0.1156 detik. Dengan demikian diperoleh rata-rata *settling time* adalah sebesar 0.3468 detik. Tetapi ini hanya terjadi pada saat *starting* awal motor induksi saja. Sedangkan pada saat *steady state* dan terjadi perubahan beban, respon kecepatan dengan DTC-FLC mengalami penurunan kecepatan yang sangat signifikan dan membutuhkan waktu untuk mencapai *steady state* lebih lama jika dibandingkan dengan DTC-SMC-GA, hampir tidak mengalami perubahan kecepatan dan waktu mencapai *steady state* lebih cepat.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil perancangan, simulasi dan analisis hasil simulasi maka dapat diperoleh kesimpulan sebagai berikut:

1. Proses pengontrolan model motor induksi DTC dengan kontroler SMC berbasis *Algoritma Genetika* dapat memberikan respon kecepatan rotor yang membentuk grafik respon seperti karakteristik sistem orde pertama. Selain itu grafik respon mampu mencapai nilai kecepatan acuan yang diberikan yaitu 1000, 1100 dan 1200 rad/m dengan rata-rata *settling time* yaitu 0,5618 detik.
2. Desain DTC-SMC berbasis *Algoritma Genetika* dapat meminimalkan fluktuasi *ripple* torsi pada saat *steady state* yaitu sebesar 2.594 Nm jika dibandingkan dengan DTC-SMC tanpa *Algoritma Genetika* fluktuasi *ripple* torsi yang dibangkitkan sebesar 3.201 Nm.
3. Penggunaan kontroler DTC-SMC berpengaruh terhadap respon fluks stator dimana pada saat *steady state* awal fluktuasi *ripple* yang dibangkitkan kecil yaitu 0.017 Wb. Nilai 0.017 Wb tersebut masih bisa diminimalisasi dengan DTC-SMC berbasis *Algoritma Genetika* yaitu sebesar 0.014 Wb.
4. Dengan digunakan kontroler DTC-SMC dan DTC-SMC berbasis *Algoritma Genetika* respon kecepatan motor induksi hampir tidak mengalami perubahan kecepatan dari referensi yang diberikan pada saat terjadi perubahan beban, dan waktu respon kecepatan dan torsi mencapai *steady state* lebih cepat. Saat diterapkan beban pada saat waktu 1.5 detik respon torsi mencapai *steady state* yaitu 1.55 detik, sedangkan pada DTC-FLC respon torsi mencapai *steady state* yaitu 1.68 detik.

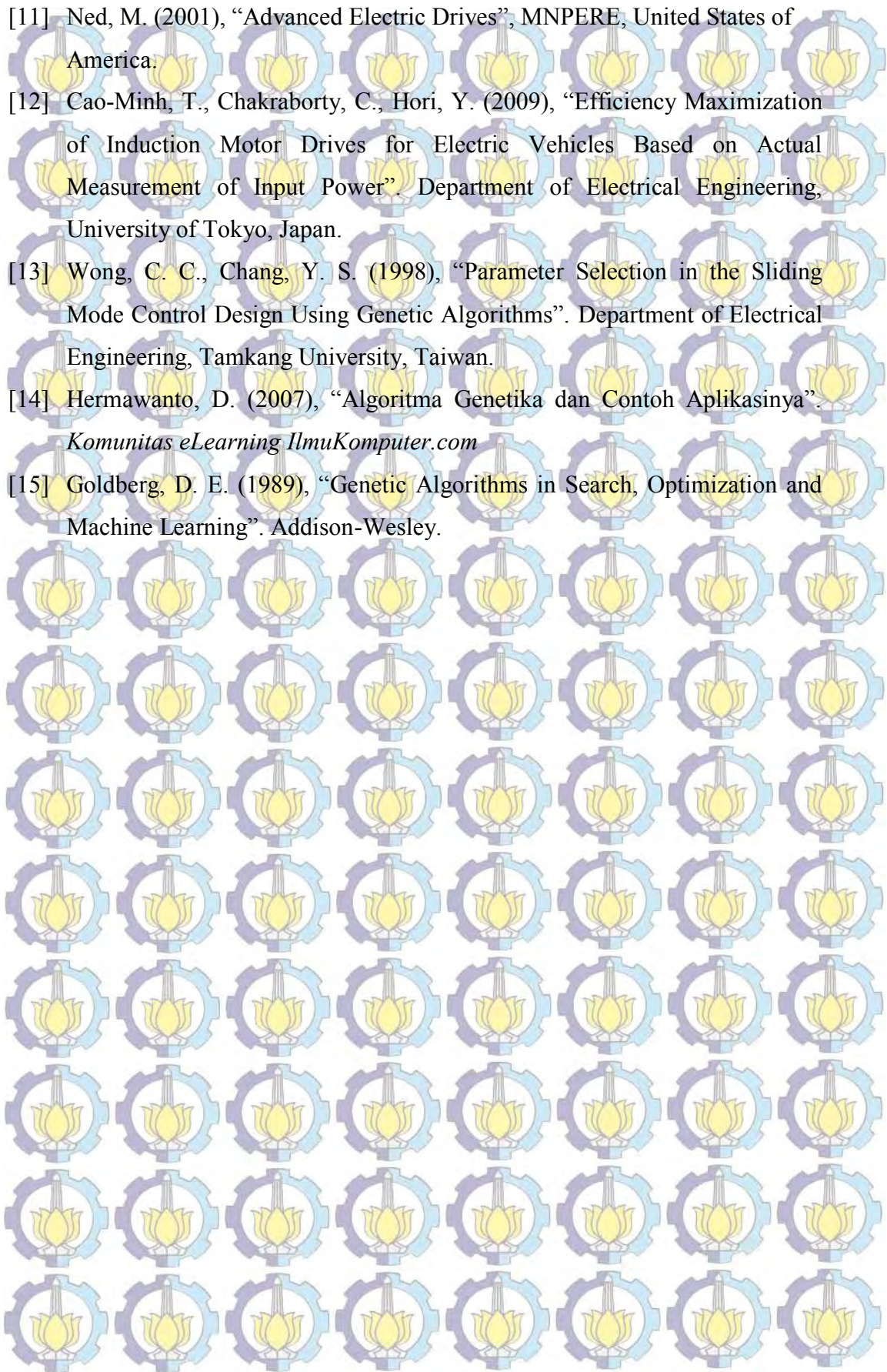
5.2 Saran

Dari hasil penelitian yang telah dilakukan, maka penulis menyarankan untuk penelitian ke depannya:

1. Karena penelitian ini hanya bersifat simulasi dari model matematis motor induksi dan komponen peralatan lainnya, maka penelitian ke depan perlu diimplementasikan dan diujicobakan langsung pada peralatan dan komponen yang real.
2. Untuk mendapatkan performa yang lebih baik lagi sistem pengaturan kecepatan motor induksi dengan kontroler SMC dapat dikombinasikan dengan *Fuzzy Logic Controller* (FLC), agar supaya respon *transient* cepat mencapai *steady state*.
3. Perlu pengkajian yang lebih mendalam dalam menerapkan Algoritma Genetika dalam optimasi parameter SMC supaya mendapatkan hasil yang lebih baik lagi.

DAFTAR PUSTAKA

- [1] Mochammad Rameli .(2014), “Bahan Kuliah Pegaturan Mesin Listrik: Motor Listrik”. Teknik Sistem Pengaturan, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya.
- [2] Mochammad Rameli. (2014), “Bahan Kuliah Pegaturan Mesin Listrik: Pengaturan Vektor Motor Induksi”. TekniK Sistem Pengaturan, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya.
- [3] Ramesh, T., Panda, K. A. (2012), “Direct Flux and Torque Control of There Phase Induction Motor Drive Using PI and Fuzzy Logic Controllers for Speed Controller for Regulator and Low Torque Ripple ”. Department of Electrical Engineering, National Institute of Tecnology, India.
- [4] Hu, F.C., Hong, B. R., Liu, H.C. (2014), “Stability analysis and PI controller tuning for a speed sensorless vector-controlled induction motor drive”, *30th Annual Conference of IEEE Inds. Elec., Society, IECON*, vol.1, 2-6 Nov, Korea.
- [5] Ahammad, T., Beig, A.R., Al-Hosani, K. (2013), “An Improved Direct Torque Control of Induction Motor with Modified Sliding Mode Control Approach”. *IEEE 2013*
- [6] Aguilar, G.M., Cortez, L. (2012), “Implementation of the Direct Torque Control (DTC) in current model, with current starting limiter”. Faculty of Sciences of the Electronics, BUAP Puebla, Mexico.
- [7] Sun, D. (2010), “Sliding Mode Direct Torque Control for Induction Motor with Robust Stator Flux observer”, *IEEE 2010 International Conference on Intelligent Computation Technology and Automation*, China.
- [8] Trzynadlowski, M. A. (2001), “Control of Induction Motors”, Academic Press, Nevada.
- [9] Robyns, B., Franscois, B., Degobert, B., Hautier, P. J. (2012), “Vector Control of Induction Machines Desentisitation and Optimisation through Fuzzy Logic”, Springer, France.
- [10] Bose, K. B. (2002), “Modern Power Electronics and AC Drives”, Prentice Hall, Knoxville.

- 
- [11] Ned, M. (2001), “Advanced Electric Drives”, MNPERE, United States of America.
- [12] Cao-Minh, T., Chakraborty, C., Hori, Y. (2009), “Efficiency Maximization of Induction Motor Drives for Electric Vehicles Based on Actual Measurement of Input Power”. Department of Electrical Engineering, University of Tokyo, Japan.
- [13] Wong, C. C., Chang, Y. S. (1998), “Parameter Selection in the Sliding Mode Control Design Using Genetic Algorithms”. Department of Electrical Engineering, Tamkang University, Taiwan.
- [14] Hermawanto, D. (2007), “Algoritma Genetika dan Contoh Aplikasinya”. *Komunitas eLearning IlmuKomputer.com*
- [15] Goldberg, D. E. (1989), “Genetic Algorithms in Search, Optimization and Machine Learning”. Addison-Wesley.

LAMPIRAN A

Embedded MATLAB Function Model Motor Induksi

```
function
[iqsd,idsd,iqrd,idrd,Te]=motind(iqs,ids,iqr,idr,vqs,vds
,vqr,vdr,we,wr)

%Parameter motor induksi
Ls =0.3829;
Lr =0.3811;
Lm =0.369;
Rs =1.77;
Rr =1.34;
P = 4;

%Matriksarusdantegangan
im=[iqs ids iqridr]';%arus
vin=[vqsvdsvqrvdr]';%tegangan

%matriks P dan Q
pe = [Rs we*Ls 0 we*Lm; -we*LsRs -we*Lm 0; 0 (we-wr)*Lm
Rr (we-wr)*Lr; -(we-wr)*Lm 0 -(we-wr)*LrRr];
qi = [Ls 0 Lm 0; 0 Ls 0 Lm; Lm 0 Lr 0; 0 Lm 0 Lr];

%Persamaanmencariturunanarus (Persamaan 3.1)
imdot=-inv(qi)*pe*im + inv(qi)*vin;

%Persamaanaturanarus
iqsd=imdot(1);
idsd=imdot(2);
iqrd=imdot(3);
idrds=imdot(4);

%Persamaanmencari torsi elektromagnetik yang
dibangkitkan
Te=(3/2)*(P/2)*Lm*( iqsd*idr - ids*iqr );
end
```


LAMPIRAN B

Mfile Algoritma Genetika Parameter SMC

```
% Algoritma Genetika Standar untuk masalah optimasi terdiri dari:  
%  
% 1. Satu populasi dengan UkPop kromosom  
% 2. Binary encoding  
% 3. Linear fitness ranking  
% 4. Roulette-wheel selection  
% 5. Pindah silang satu titik potong  
% 6. Probabilitas pindah silang pc dan probabilitas mutasi pmut  
% bernilai tetap  
% 7. Elitisme, satu atau dua buah kopi dari individu bernilai  
% fitness tertinggi  
% 8. Generational replacement: mengganti semua individu dengan  
% individu baru  
%
```

```
Nvar      = 1;          % Jumlah variabel pada fungsi yang  
dioptimasi  
Nbit      = 10;         % Jumlah bit yang mengkodekan satu  
variabel  
JumGen    = Nbit*Nvar;  % Jumlah gen dalam kromosom  
Rb        = 0;          % Batas bawah interval  
Ra        = 2;          % Batas atas interval  
  
UkPop     = 10;         % Jumlah kromosom dalam populasi  
Psilang   = 0.8;        % Probabilitas pindah silang  
Pmutasi   = 0.005;      % Probabilitas mutasi  
MaxG      = 10;         % Jumlah generasi  
  
Fthreshold = 20;        % Threshold untuk nilai Fitness  
Bgraf     = Fthreshold; % Untuk menangani tampilan grafis  
  
% Inisialisasi grafis 2D  
hfig = figure;  
hold on  
title('Optimasi fungsi kompleks menggunakan AG standar')  
set(hfig, 'position', [50,50,600,400]);  
set(hfig, 'DoubleBuffer', 'on');  
axis([1 MaxG 0 Bgraf]);  
hbestplot = plot(1:MaxG, zeros(1,MaxG));  
htext1 = text(0.6*MaxG, 0.75*Bgraf, sprintf('Fitness terbaik: %9.8f',  
0.0));  
htext2 = text(0.6*MaxG, 0.70*Bgraf, sprintf('X : %9.4f', 0.0));  
htext3 = text(0.6*MaxG, 0.65*Bgraf, sprintf('X2 : %9.4f', 0.0));  
htext4 = text(0.6*MaxG, 0.60*Bgraf, sprintf('X3 : %9.8f', 0.0));  
htext5 = text(0.6*MaxG, 0.55*Bgraf, sprintf('X4 : %9.8f', 0.0));  
htext6 = text(0.6*MaxG, 0.50*Bgraf, sprintf('X5 : %9.8f', 0.0));  
htext7 = text(0.6*MaxG, 0.45*Bgraf, sprintf('X6 : %9.8f', 0.0));  
htext8 = text(0.6*MaxG, 0.40*Bgraf, sprintf('X7 : %9.8f', 0.0));  
htext9 = text(0.6*MaxG, 0.35*Bgraf, sprintf('X8 : %9.8f', 0.0));  
htext10 = text(0.6*MaxG, 0.30*Bgraf, sprintf('X9 : %9.8f', 0.0));
```



```

%htext11 = text(0.6*MaxG,0.25*Bgraf,sprintf('X10: %9.8f', 0.0));
htext12 = text(0.6*MaxG,0.20*Bgraf,sprintf('Nilai maksimum: %9.8f',
0.0));
htext13 = text(0.6*MaxG,0.15*Bgraf,sprintf('Ukuran
populasi: %3.0f', 0.0));
htext14 = text(0.6*MaxG,0.10*Bgraf,sprintf('Probabilitas
mutasi: %4.3f', 0.0));
htext15 = text(0.6*MaxG,0.05*Bgraf,sprintf('Jumlah
bit/variabel: %4.3f', 0.0));
xlabel('Generasi');
ylabel('Fitness terbaik');
hold off
drawnow;

% Inisialisasi populasi
Populasi = InisialisasiPopulasi(UkPop,JumGen);

% Loop evolusi
for generasi=1:MaxG,
    x = DekodekanKromosom(Populasi(1,:),Nvar,Nbit,Ra,Rb);
    kak=2.5+5*x/1023;
    assignin('base','ori',12);
    assignin('base','kak',kak);
    sim('SMCCOBATES');
    sumer=sum(jekwad)/150;
    Fitness(1) = 1/sumer;
    MaxF = Fitness(1);
    MinF = Fitness(1);
    IndeksIndividuTerbaik = 1;
    for ii=2:UkPop,
        Kromosom = Populasi(ii,:);
        x = DekodekanKromosom(Kromosom,Nvar,Nbit,Ra,Rb);
        kak=2.5+4*x/1023;
        assignin('base','ori',12);
        assignin('base','kak',kak);
        sim('SMCCOBATES');
        sumer=sum(jekwad)/150;
        Fitness(ii) = 1/sumer;
        if (Fitness(ii) > MaxF),
            MaxF = Fitness(ii);
            IndeksIndividuTerbaik = ii;
            BestX = x;
        end
        if (Fitness(ii) < MinF),
            MinF = Fitness(ii);
        end
    end
end

% Penanganan grafis 2D
plotvector = get(hbestplot,'YData');
plotvector(generasi) = MaxF;
set(hbestplot,'YData',plotvector);
set(htext1,'String',sprintf('Fitness terbaik: %9.8f', MaxF));
set(htext2,'String',sprintf('X : %9.8f', BestX(1)));
%set(htext3,'String',sprintf('X2 : %9.8f', BestX(2)));
%set(htext4,'String',sprintf('X3 : %9.8f', BestX(3)));
%set(htext5,'String',sprintf('X4 : %9.8f', BestX(4)));
%set(htext6,'String',sprintf('X5 : %9.8f', BestX(5)));

```



```

    %set(htext7,'String',sprintf('X6 : %9.8f', BestX(6)));
    %set(htext8,'String',sprintf('X7 : %9.8f', BestX(7)));
    %set(htext9,'String',sprintf('X8 : %9.8f', BestX(8)));
    %set(htext10,'String',sprintf('X9 : %9.8f', BestX(9)));
    set(htext12,'String',sprintf('Nilai maksimum: %9.8f',
MaxF));
    set(htext13,'String',sprintf('Ukuran populasi: %3.0f', UkPop));
    set(htext14,'String',sprintf('Probabilitas mutasi: %4.3f',
Pmutasi));
    set(htext15,'String',sprintf('Jumlah bit/variabel: %2.0f',
Nbit));
    drawnow

    if MaxF >= Fthreshold,
        break;
    end

    TempPopulasi = Populasi;

    % Elitisme:
    % - Buat satu kopi kromosom terbaik jika ukuran populasi ganjil
    % - Buat dua kopi kromosom terbaik jika ukuran populasi genap
    if mod(UkPop,2)==0, % ukuran populasi genap
        IterasiMulai = 3;
        TempPopulasi(1,:) = Populasi(IndeksIndividuTerbaik,:);
        TempPopulasi(2,:) = Populasi(IndeksIndividuTerbaik,:);
    else % ukuran populasi ganjil
        IterasiMulai = 2;
        TempPopulasi(1,:) = Populasi(IndeksIndividuTerbaik,:);
    end

    LinearFitness = LinearFitnessRanking(UkPop,Fitness,MaxF,MinF);

    % Roulette-wheel selection dan pindah silang
    for jj=IterasiMulai:2:UkPop,
        IP1 = RouletteWheel(UkPop,LinearFitness);
        IP2 = RouletteWheel(UkPop,LinearFitness);
        if (rand < Psilang),
            Anak =
PindahSilang(Populasi(IP1,:),Populasi(IP2,:),JumGen);
            TempPopulasi(jj,:) = Anak(1,:);
            TempPopulasi(jj+1,:) = Anak(2,:);
        else
            TempPopulasi(jj,:) = Populasi(IP1,:);
            TempPopulasi(jj+1,:) = Populasi(IP2,:);
        end
    end

    % Mutasi dilakukan pada semua kromosom
    for kk=IterasiMulai:UkPop,
        TempPopulasi(kk,:) =
Mutasi(TempPopulasi(kk,:),JumGen,Pmutasi);
    end

    % Generational Replacement: mengganti semua kromosom sekaligus
    Populasi = TempPopulasi
end

```



```

% Mendekodekan kromosom yang berisi bilangan biner menjadi
individu x yang
% bernilai real dalam interval yang ditentukan [Ra,Rb]
%
% Masukan
% Kromosom : kromosom, matriks berukuran 1 x JumGen
% Nvar : jumlah variabel
% Nbit : jumlah bit yang mengkodekan satu variabel
% Ra : batas atas interval
% Rb : batas bawah interval
function x = DekodekanKromosom(Kromosom,Nvar,Nbit,Ra,Rb)

for ii=1:Nvar,
    x(ii) = 0;
    for jj=1:Nbit,
        x(ii) = x(ii) + Kromosom((ii-1)*Nbit+jj)*2^(-jj);
    end
    x(ii) = Rb + (Ra-Rb)*x(ii);
end
% Mengevaluasi individu sehingga didapatkan nilai fitness-nya
%
% Masukan
% x : individu
%
% Keluaran
% fitness : nilai fitness
%
function fitness = EvaluasiIndividu(x)

a=-5;
b=-7;
c=9;

fitness = a*x*x+b*x+c;

% Membangkitkan sejumlah UkPop kromosom, masing-masing kromosom
berisi bilangan
% biner (0 dan 1) sejumlah JumGen
%
% Masukan
% UkPop : ukuran populasi atau jumlah kromosom dalam populasi
% JumGen: jumlah gen dalam kromosom
%
% Keluaran
% Populasi : kumpulan kromosom, matriks berukuran UkPop x JumGen
%
function Populasi = InisialisasiPopulasi(UkPop,JumGen)

Populasi = fix(2*rand(10,10));

% Men-skala-kan nilai fitness ke dalam ranking sehingga diperoleh
% nilai-nilai fitness baru yang berada dalam rentang [MaxF,MinF]
%
% Masukan
% UkPop : ukuran populasi atau jumlah kromosom dalam populasi
% Fitness: nilai fitness, matriks ukuran 1 x UkPop

```



```

% MaxF : nilai fitness maximum
% MinF : nilai fitness minimum
% Keluaran
% LFR : Linear Fitness Ranking
function LFR = LinearFitnessRanking(UkPop,Fitness,MaxF,MinF)

% SF berisi nilai fitness yang terurut dari kecil ke besar
(ascending)
% IndF berisi index dari nilai fitness yang menyatakan nomor urut
kromosom
[SF,IndF] = sort(Fitness);

% LinearFitness = nilai fitness baru hasil pen-skala-an
for rr=1:UkPop,
    LFR(IndF(UkPop-rr+1)) = MaxF-(MaxF-MinF)*((rr-1)/(UkPop-1));
end

% Mutasi gen dengan probabilitas sebesar Pmutasi
% Gen-gen yang terpilih diubah nilainya: 0 menjadi 1, dan 1
menjadi 0
%
% Masukan
% Kromosom : kromosom, matriks berukuran 1 x JumGen
% JumGen : jumlah gen
% Pmutasi : Probabilitas mutasi
%
% Keluaran
% MutKrom : kromosom hasil mutasi, matriks berukuran 1 x JumGen
function MutKrom = Mutasi(Kromosom,JumGen,Pmutasi);

MutKrom = Kromosom;
for ii=1:JumGen,
    if (rand < Pmutasi),
        if Kromosom(ii)==0,
            MutKrom(ii) = 1;
        else
            MutKrom(ii) = 0;
        end
    end
end
end

```


BIOGRAFI PENULIS



M. Nur Faizi lahir di Bandul, Meranti, Riau, 2 februari 1987, dan telah menempuh pendidikan formal yaitu di SDN 01 Bandul, SLTPN 01 Bandul, dan SMAN 2 Bengkalis lulus pada tahun 2005, kemudian melanjutkan jenjang pendidikan Diploma 3 di Politeknik Negeri Bengkalis pada tahun 2008 dan lulus pada tahun 2011. Selanjutnya Penulis meneruskan pendidikan Diploma 4 di Politeknik Elektronika Negeri Surabaya dengan konsentrasi bidang keahlian teknik elektro industri pada tahun 2011 dan lulus pada tahun 2013. Di tahun 2013, penulis berkesempatan untuk melanjutkan pendidikan Pascasarjana strata-2 di Institut Teknologi Sepuluh Nopember dengan konsentrasi bidang keahlian teknik sistem pengaturan. Pada bulan januari 2016 penulis mengikuti seminar dan ujian tesis sebagai salah satu syarat untuk memperoleh gelar Master Teknik (M.T). Jika ingin berkomunikasi lebih lanjut bisa hubungi email : emnoer_faizi@yahoo.com.

[Halaman ini sengaja dikosongkan]

